Logical Methods in Computer Science
Vol. 6 (2:4) 2010, pp. 1–33
www.lmcs-online.org

Submitted Dec. 18, 2009
Published Jun. 23, 2010

ANALYTIC TABLEAUX FOR SIMPLE TYPE THEORY AND ITS FIRST-ORDER FRAGMENT

CHAD E. BROWN [a] AND GERT SMOLKA [b]

[a,b] Saarland University
e-mail address: {cebrown,smolka}@ps.uni-saarland.de

Abstract. We study simple type theory with primitive equality (STT) and its first-order fragment EFO, which restricts equality and quantification to base types but retains lambda abstraction and higher-order variables. As deductive system we employ a cut-free tableau calculus. We consider completeness, compactness, and existence of countable models. We prove these properties for STT with respect to Henkin models and for EFO with respect to standard models. We also show that the tableau system yields a decision procedure for three EFO fragments.

1. Introduction

Church's type theory [16] is a basic formulation of higher-order logic. Henkin [18] found a natural class of models for which Church's Hilbert-style proof system turned out to be complete. Equality, originally expressed with higher-order quantification, was later identified as the primary primitive of the theory [19, 3, 1]. In this paper we consider simple type theory with primitive equality but without descriptions or choice. We call this system STT for simple type theory. The semantics of STT is given by Henkin models with equality.

Modern proof theory started with Gentzen's [17] invention of a cut-free sequent calculus for first-order logic. While Gentzen proved a cut-elimination theorem for his calculus, Smullyan [25] found an elegant technique (abstract consistency classes) for proving the completeness of cut-free first-order calculi. Smullyan [25] found it advantageous to work with a refutation-oriented variant of Gentzen's sequent calculi [17] known as tableau calculi [10, 20, 25].

The development of complete cut-free proof systems for simple type theory turned out to be hard. In 1953, Takeuti [30] introduced a sequent calculus for a version of simple type theory without primitive equality and conjectured that cut elimination holds for this calculus. Gentzen's [17] inductive proof of cut-elimination for first-order sequent calculi does not generalize to the higher-order case since instances of formulas may be more complex than the formula itself. Moreover, Henkin's [18] completeness proof cannot be adapted for cut-free systems. Takeuti's conjecture was answered positively by Tait [27] for second-order

1998 ACM Subject Classification: F.4.1,I.2.3.

Key words and phrases: higher-order logic, simple type theory, first-order logic, tableaux, completeness, cut-elimination, decision procedures.

DOI:10.2168/LMCS-6 (2:4) 2010

logic, by Takahashi [28] and Prawitz [24] for higher-order logic without extensionality, and by Takahashi [29] for higher-order logic with extensionality. Building on the possible-values technique of Takahashi [28] and Prawitz [24], Takeuti [31] finally proves Henkin completeness of a cut-free sequent calculus with extensionality.

The first cut-elimination result for a calculus similar to Church's type theory was obtained by Andrews [2] in 1971. Andrews considers elementary type theory (Church's type theory without equality, extensionality, infinity, and choice) and proves that a cut-free sequent calculus is complete relative to a Hilbert-style proof system. Andrews' proof employs both the possible-values technique [28, 24] and the abstract consistency technique [25]. In 2004 Benzmüller, Brown and Kohlhase [7] gave a completeness proof for an extensional cut-free sequent calculus. The constructions in [7] also employ abstract consistency and possible values.

None of the cut-free calculi discussed above has equality as a primitive. Following Leibniz, one can define equality of a and b to hold whenever a and b satisfy the same properties. While this yields equality in standard models (full function spaces), there are Henkin models where this is not the case as was shown by Andrews [3]. A particularly disturbing fact about the model Andrews constructs is that while it is extensional (indeed, it is a Henkin model), it does not satisfy a formula corresponding to extensionality (formulated using Leibniz equality). In [3] Andrews gives a definition of a *general model* which is essentially a Henkin model with equality. This notion of a general model was generalized to include non-extensional models in [6] and a condition called property $\mathfrak{q}$ was explicitly included to ensure Leibniz equality is the same as semantic equality. The constructions of Prawitz, Takahashi, Andrews and Takeuti described above do not produce models guaranteed to satisfy property $\mathfrak{q}$. A similar generalization of Henkin models to non-extensional models is given by Muskens [23] but without a condition like property $\mathfrak{q}$. Muskens uses the Prawitz-Takahashi method to prove completeness of a cut-free sequent calculus for a formulation of elementary type theory via a model existence theorem, again producing a model in which Leibniz equality may not be the same as semantic equality. The models constructed in [6] do satisfy property $\mathfrak{q}$, as do the models constructed in [7].

In addition to the model-theoretic complication, defined equality also destroys the cut-freeness of a proof system. As shown in [8] any use of Leibniz equality to say two terms are equal provides for the simulation of cut.[1] Hence calculi that define equality as Leibniz equality cannot claim to provide cut-free equational reasoning. In the context of resolution, Benzmüller gives serious consideration to primitive equality and its relationship to Leibniz equality in his 1999 doctoral thesis [4] (see also [5]). The completeness proofs there are relative to an assumption that corresponds to cut.

The first completeness proof for a cut-free proof system for extensional simple type theory with primitive equality relative to Henkin models was given by Brown in his 2004 doctoral thesis [12] (later published as a book [13]). Brown proves the Henkin completeness of a novel one-sided sequent calculus with primitive equality. His model construction starts with Andrews' [2] non-extensional possible-values relations and then obtains a structure isomorphic to a Henkin model by taking a quotient with respect to a partial equivalence relation. Finally, abstract consistency classes [25, 2] are used to obtain the completeness result. The equality-based decomposition rules of Brown's sequent calculus have commonalities with the unification rules of the systems of Kohlhase [22] and Benzmüller [5]. Note,

[1] From a Leibniz formula of the form $\forall p.ps \to pt$ one can easily infer $u \to u$ for any formula u, and then use u as a formula introduced by cut.

however, that the completeness proofs of Kohlhase and Benzmüller assume the presence of cut.

In this paper we improve and simplify Brown's result [13]. For the proof system we switch to a cut-free tableau calculus $\mathcal{T}$ that employs an abstract normalization operator. With the normalization operator we hide the details of lambda conversion from the tableau calculus and most of the completeness proof. For the completeness proof we use the new notion of a value system to directly construct surjective Henkin models. Value systems are logical relations [26] providing a relational semantics for simply-typed lambda calculus. The inspiration for value systems came from the possible-values relations used in [13, 15, 14]. In contrast to Henkin models, which obtain values for terms by induction on terms, value systems obtain values for terms by induction on types. Induction on types, which is crucial for our proofs, has the advantage of hiding the presence of the lambda binder. As a result, only a single lemma of our completeness proof deals explicitly with lambda abstractions and substitutions.

Once we have established the results for STT, we turn to its first-order fragment EFO (for extended first-order), which restricts equality and quantification to base types but retains lambda abstraction and higher-order variables. EFO contains the usual first-order formulas but also contains formulas that are not first-order in the traditional sense. For instance, a formula $p(\lambda x.\neg fx)$ is EFO even though the predicate p is applied to a λ-abstraction and the negation appears embedded in a nontrivial way. We sharpen the results for STT by proving that they hold for EFO with respect to standard models and for a constrained rule for the universal quantifier (first published in [14]).

Finally, we consider three decidable fragments of EFO: the lambda-free fragment, the pure fragment (disequations between simply typed λ-terms not involving logic), and the Bernays-Schönfinkel-Ramsey fragment. For each of these fragments, decidability follows from termination of the tableau calculus for EFO (first published in [15] and [14]).

2. Basic Definitions

We assume a countable set of *base types* (β). *Types* (σ, τ, μ) are defined inductively: (1) every base type is a type; (2) if σ and τ are types, then $\sigma\tau$ is a type. We assume a countable set of *names* (x, y), where every name comes with a unique type, and where for every type there are infinitely many names of this type.[2] *Terms* (s, t, u, v) are defined inductively: (1) every name is a term; (2) if s is a term of type $\tau\mu$ and t is a term of type τ, then st is a term of type μ; (3) if x is a name of type σ and t is a term of type τ, then $\lambda x.t$ is a term of type $\sigma\tau$. We write $s : \sigma$ to say that s is a term of type σ. Moreover, we write Λ_σ for the set of all terms of type σ. We assume that the set of types and the set of terms are disjoint.

A *frame* is a function $\mathcal{D}$ that maps every type to a nonempty set such that $\mathcal{D}(\sigma\tau)$ is a set of total functions from $\mathcal{D}\sigma$ to $\mathcal{D}\tau$ for all types σ, τ (i.e., $\mathcal{D}(\sigma\tau) \subseteq (\mathcal{D}\sigma \to \mathcal{D}\tau)$). An *assignment* into a frame $\mathcal{D}$ is a function $\mathcal{I}$ that extends $\mathcal{D}$ (i.e., $\mathcal{D} \subseteq \mathcal{I}$) and maps every name $x : \sigma$ to an element of $\mathcal{D}\sigma$ (i.e., $\mathcal{I}x \in \mathcal{D}\sigma$). If $\mathcal{I}$ is an assignment into a frame $\mathcal{D}$, $x : \sigma$ is a name, and $a \in \mathcal{D}\sigma$, then $\mathcal{I}^x_a$ denotes the assignment into $\mathcal{D}$ that agrees everywhere with $\mathcal{I}$ but possibly on x where it yields a. For every frame $\mathcal{D}$ we define a function $\hat{}$ that for

[2] Later we will partition names into variables and logical constants.

every assignment $\mathcal{I}$ into $\mathcal{D}$ yields a function $\hat{\mathcal{I}}$ that for some terms $s : \sigma$ returns an element of $\mathcal{D}\sigma$. The definition is by induction on terms.

$$\begin{aligned}\hat{\mathcal{I}}x &:= \mathcal{I}x \\ \hat{\mathcal{I}}(st) &:= fa && \text{if } \hat{\mathcal{I}}s = f \text{ and } \hat{\mathcal{I}}t = a \\ \hat{\mathcal{I}}(\lambda x.s) &:= f && \text{if } \lambda x.s : \sigma\tau,\ f \in \mathcal{D}(\sigma\tau), \text{ and } \forall a \in \mathcal{D}\sigma : \widehat{\mathcal{I}^x_a}s = fa\end{aligned}$$

We call $\hat{\mathcal{I}}$ the *evaluation function* of $\mathcal{I}$. The evaluation function may be partial since in the last clause of the definition even assuming there is some function f such that $\widehat{\mathcal{I}^x_a}s = fa$ for every $a \in \mathcal{D}\sigma$, this f may not be in $\mathcal{D}(\sigma\tau)$. In such a case, $\hat{\mathcal{I}}$ will not be defined on $\lambda x.s$. Of course, in such a case $\hat{\mathcal{I}}$ will also not be defined on a term of the form $(\lambda x.s)t$ since the second clause of the definition will fail. An *interpretation* is an assignment whose evaluation function is defined on all terms. An assignment $\mathcal{I}$ is *surjective* if for every type σ and every value $a \in \mathcal{I}\sigma$ there exists a term $s : \sigma$ such that $\hat{\mathcal{I}}s = a$.

Proposition 2.1. *Let $\mathcal{I}$ be an interpretation, $x : \sigma$, and $a \in \mathcal{I}\sigma$. Then $\mathcal{I}^x_a$ is an interpretation.*

Proposition 2.2. *If $\mathcal{I}$ is a surjective interpretation, then $\mathcal{I}\sigma$ is a countable set for every type σ.*

A *standard frame* is a frame $\mathcal{D}$ such that $\mathcal{D}(\sigma\tau) = (\mathcal{D}\sigma \to \mathcal{D}\tau)$ for all types σ, τ. A *standard interpretation* is an assignment into a standard frame. Note that every standard interpretation is, in fact, an interpretation.

We assume a *normalization operator* $[\cdot]$ that provides for lambda conversion. The normalization operator $[\cdot]$ must be a type preserving total function from terms to terms. We call $[s]$ the *normal form of* s and say that s is *normal* if $[s] = s$. One possible normalization operator is a function that for every term s return a β-normal term that can be obtained from s by β-reduction. We will not commit to a particular normalization operator but state explicitly the properties we require for our results. To start, we require the following properties:

N1 : $[[s]] = [s]$
N2 : $[[s]t] = [st]$
N3 : $[xs_1 \dots s_n] = x[s_1] \dots [s_n]$ if $xs_1 \dots s_n : \beta$ and $n \geq 0$
N4 : $\hat{\mathcal{I}}[s] = \hat{\mathcal{I}}s$ if $\mathcal{I}$ is an interpretation

Proposition 2.3. *$xs_1 \dots s_n : \beta$ is normal iff $s_1, \dots, s_n$ are normal.*

For the proofs of Lemma 3.3 and Theorem 3.4 we need further properties of the normalization operator that can only be expressed with substitutions. A *substitution* is a type preserving partial function from names to terms. If θ is a substitution, x is a name, and s is a term that has the same type as x, we write θ^x_s for the substitution that agrees everywhere with θ but possibly on x where it yields s. We assume that every substitution θ can be extended to a type preserving total function $\hat{\theta}$ from terms to terms such that the following conditions hold:

S1 : $\hat{\theta}x = \text{if } x \in \text{Dom}\,\theta \text{ then } \theta x \text{ else } x$
S2 : $\hat{\theta}(st) = (\hat{\theta}s)(\hat{\theta}t)$
S3 : $[(\hat{\theta}(\lambda x.s))t] = [\widehat{\theta^x_t}s]$
S4 : $[\hat{\emptyset}s] = [s]$

Note that $\emptyset$ (the empty set) is the substitution that is undefined on every name.

3. Value Systems

We introduce value systems as a tool for constructing surjective interpretations. Value systems are logical relations inspired by the possible-values relations used in [13, 14, 15].

A *value system* is a function $\triangleright$ that maps every base type β to a binary relation $\triangleright_\beta$ such that $\mathrm{Dom}\,(\triangleright_\beta) \subseteq \Lambda_\beta$ and $s \triangleright_\beta a$ iff $[s] \triangleright_\beta a$. For every value system $\triangleright$ we define by induction on types:

$$\begin{aligned} \mathcal{D}\sigma &:= \mathrm{Ran}\,(\triangleright_\sigma) \\ \triangleright_{\sigma\tau} &:= \{\, (s,f) \in \Lambda_{\sigma\tau} \times (\mathcal{D}\sigma \to \mathcal{D}\tau) \mid \forall (t,a) \in \triangleright_\sigma \colon (st, fa) \in \triangleright_\tau \,\} \end{aligned}$$

Note that $\mathcal{D}(\sigma\tau) \subseteq (\mathcal{D}\sigma \to \mathcal{D}\tau)$ for all types $\sigma\tau$. We usually drop the type index in $s \triangleright_\sigma a$ and read $s \triangleright a$ as s can be a or a is a *possible value* for s.

Proposition 3.1. *For every value system: $s \triangleright_\sigma a$ iff $[s] \triangleright_\sigma a$.*

Proof. By induction on σ. For base types the claim holds by the definition of value systems. Let $\sigma = \tau\mu$. For all $s \in \Lambda_\sigma$, $t \in \Lambda_\tau$, $a \in \mathcal{D}\tau \to \mathcal{D}\mu$, and $b \in \mathcal{D}\tau$,

$$st \triangleright_\mu ab \text{ iff } [st] \triangleright_\mu ab \text{ iff } [[s]t] \triangleright_\mu ab \text{ iff } [s]t \triangleright_\mu ab$$

by the inductive hypothesis and N2. Hence $s \triangleright_\sigma a$ iff $[s] \triangleright a$. □

A value system $\triangleright$ is *functional* if $\triangleright_\beta$ is a functional relation for every base type β. (That is, for each $s \in \Lambda_\beta$ there is at most one b such that $s \triangleright b$.)

Proposition 3.2. *If $\triangleright$ is functional, then $\triangleright_\sigma$ is a functional relation for every type σ.*

Proof. By induction on σ. For $\sigma = \beta$, the claim is trivial. Let $\sigma = \tau\mu$ and $s \triangleright_{\tau\mu} f, g$. We show $f = g$. Let $a \in \mathcal{D}\tau$. Then $t \triangleright_\tau a$ for some t. Now $st \triangleright_\mu fa, ga$. By inductive hypothesis $fa = ga$. □

A value system $\triangleright$ is *total* if $x \in \mathrm{Dom}\,\triangleright_\sigma$ for every name $x : \sigma$. An assignment $\mathcal{I}$ is *admissible* for a value system $\triangleright$ if $\mathcal{I}\sigma = \mathcal{D}\sigma$ for all types σ and $x \triangleright \mathcal{I}x$ for all names x. (Recall that $\triangleright$ is used to define $\mathcal{D}$.) Note that every total value system has admissible assignments. We will show that admissible assignments are interpretations that evaluate terms to possible values.

Lemma 3.3. *Let $\mathcal{I}$ be an assignment that is admissible for a value system $\triangleright$ and θ be a substitution such that $\theta x \triangleright \mathcal{I}x$ for all $x \in \mathrm{Dom}\,\theta$. Then $s \in \mathrm{Dom}\,\hat{\mathcal{I}}$ and $\hat{\theta}s \triangleright \hat{\mathcal{I}}s$ for every term s.*

Proof. By induction on s. Let s be a term. Case analysis.

$s = x$. The claim holds by assumption and S1.

$s = tu$. Then $t \in \mathrm{Dom}\,\hat{\mathcal{I}}$, $\hat{\theta}t \triangleright \hat{\mathcal{I}}t$, $u \in \mathrm{Dom}\,\hat{\mathcal{I}}$, and $\hat{\theta}u \triangleright \hat{\mathcal{I}}u$ by inductive hypothesis. Thus $s \in \mathrm{Dom}\,\hat{\mathcal{I}}$ and $\hat{\theta}s = (\hat{\theta}t)(\hat{\theta}u) \triangleright (\hat{\mathcal{I}}t)(\hat{\mathcal{I}}u) = \hat{\mathcal{I}}s$ using S2.

$s = \lambda x.t$, $x : \sigma$ and $t : \tau$. We need to prove $s \in \mathrm{Dom}\,\hat{\mathcal{I}}$ and $\hat{\theta}s \triangleright \hat{\mathcal{I}}s$. First we prove

$$t \in \mathrm{Dom}\,\widehat{\mathcal{I}^x_a} \text{ and } (\hat{\theta}s)u \triangleright \widehat{\mathcal{I}^x_a}t \text{ whenever } u \triangleright_\sigma a. \tag{3.1}$$

Let $u \rhd_\sigma a$. By inductive hypothesis we have $t \in \mathrm{Dom}\,\widehat{\mathcal{I}^x_a}$ and $\widehat{\theta^x_u}t \rhd \widehat{\mathcal{I}^x_a}t$. Now $[(\hat{\theta}s)u] = [\widehat{\theta^x_u}t] \rhd \widehat{\mathcal{I}^x_a}t$ using S3. Using Proposition 3.1 we conclude (3.1) holds.

By definition of $\mathcal{D}\sigma$ for every $a \in \mathcal{D}\sigma$ there is a u such that $u \rhd a$. Using this and (3.1) we know $t \in \mathrm{Dom}\,\widehat{\mathcal{I}^x_a}$ for every $a \in \mathcal{D}\sigma$. Let $f : \mathcal{D}\sigma \to \mathcal{D}\tau$ be defined by $fa = \widehat{\mathcal{I}^x_a}t$ for each $a \in \mathcal{I}\sigma$. For all $u \rhd_\sigma a$ we have $(\hat{\theta}s)u \rhd fa$ by (3.1). Hence $\hat{\theta}s \rhd f$. This implies $f \in \mathcal{D}(\sigma\tau)$, $s \in \mathrm{Dom}\,\hat{\mathcal{I}}$, $\hat{\mathcal{I}}s = f$ and $\hat{\theta}s \rhd \hat{\mathcal{I}}s$ as desired. □

Theorem 3.4. *Let $\mathcal{I}$ be an assignment that is admissible for a value system $\rhd$. Then $\mathcal{I}$ is an interpretation such that $s \rhd \hat{\mathcal{I}}s$ for all terms s. Furthermore, $\mathcal{I}$ is surjective if $\rhd$ is functional.*

Proof. Follows from Lemma 3.3 with Proposition 3.1 and S4. To prove the second claim, let $a \in \mathcal{D}\sigma$ be given. By definition of $\mathcal{D}$ there is some s such that $s \rhd a$. Since $s \rhd \hat{\mathcal{I}}s$ we know $\hat{\mathcal{I}}s = a$ by Proposition 3.2. □

4. Simple Type Theory

We now define the terms and semantics of simple type theory (*STT*). We fix a base type o for the truth values and a name $\neg : oo$ for negation. Moreover, we fix for every type σ a name $=_\sigma : \sigma\sigma o$ for the identity predicate for σ. An assignment $\mathcal{I}$ is *logical* if $\mathcal{I}o = \{0, 1\}$, $\mathcal{I}(\neg)$ is the negation function and $\mathcal{I}(=_\sigma)$ is the identity predicate for σ. We refer to the base types different from o as *sorts*, to the names $\neg$ and $=_\sigma$ as *logical constants*, and to all other names as *variables*. From now on x will range over variables. Moreover, c will range over logical constants and α will range over sorts.

A *formula* is a term of type o. We employ infix notation for formulas obtained with $=_\sigma$ and often write *equations* $s =_\sigma t$ without the type index. We write $s \neq t$ for $\neg(s{=}t)$ and speak of a *disequation*. Note that quantified formulas $\forall x.s$ can be expressed as equations $(\lambda x.s) = (\lambda x.x = x)$.

A logical interpretation $\mathcal{I}$ *satisfies* a formula s if $\hat{\mathcal{I}}s = 1$. A *model* of a set of formulas A is a logical interpretation that satisfies every formula $s \in A$. A set of formulas is *satisfiable* if it has a model.

5. Tableau Calculus

We now give a deductive calculus for STT. A *branch* is a set of normal formulas. The *tableau calculus* $\mathcal{T}$ operates on finite branches and employs the rules shown in Figure 1. The side condition "x fresh" of rule $\mathcal{T}_{\text{FE}}$ requires that x does not occur free in the branch the rule is applied to. We say a branch A is *closed* if $x, \neg x \in A$ for some variable $x : o$ or if $x \neq_\iota x \in A$ for some variable $x : \iota$. Note that A is closed if and only if either the $\mathcal{T}_{\text{MAT}}$ or $\mathcal{T}_{\text{DEC}}$ rule applies with $n = 0$. We impose the following restrictions:

(1) We only admit rule instances $A/A_1 \ldots A_n$ where A is not closed.
(2) $\mathcal{T}_{\text{FE}}$ can only be applied to a disequation $(s{\neq}t) \in A$ if there is no variable x such that $([sx] \neq [tx]) \in A$.

The set of *refutable branches* is defined inductively: if $A/A_1 \ldots A_n$ is an instance of a rule of $\mathcal{T}$ and $A_1, \ldots, A_n$ are refutable, then A is refutable. Note that the base cases of this inductive definition are when $n = 0$. The rules where n may be 0 are $\mathcal{T}_{\text{MAT}}$ and $\mathcal{T}_{\text{DEC}}$. Figure 2 shows a refutation in $\mathcal{T}$.

$$\mathcal{T}_{\neg\neg}\ \frac{\neg\neg s}{s} \qquad \mathcal{T}_{\mathrm{BQ}}\ \frac{s =_o t}{s\,,\, t \ \mid\ \neg s\,,\, \neg t} \qquad \mathcal{T}_{\mathrm{BE}}\ \frac{s \neq_o t}{s\,,\, \neg t \ \mid\ \neg s\,,\, t}$$

$$\mathcal{T}_{\mathrm{FQ}}\ \frac{s =_{\sigma\tau} t}{[su] = [tu]}\ u : \sigma \text{ NORMAL} \qquad \mathcal{T}_{\mathrm{FE}}\ \frac{s \neq_{\sigma\tau} t}{[sx] \neq [tx]}\ x : \sigma \text{ FRESH}$$

$$\mathcal{T}_{\mathrm{MAT}}\ \frac{xs_1 \dots s_n\,,\, \neg xt_1 \dots t_n}{s_1 \neq t_1 \mid \cdots \mid s_n \neq t_n}\ n \geq 0 \qquad \mathcal{T}_{\mathrm{DEC}}\ \frac{xs_1 \dots s_n \neq_\alpha xt_1 \dots t_n}{s_1 \neq t_1 \mid \cdots \mid s_n \neq t_n}\ n \geq 0$$

$$\mathcal{T}_{\mathrm{CON}}\ \frac{s =_\alpha t\,,\, u \neq_\alpha v}{s \neq u\,,\, t \neq u \mid s \neq v\,,\, t \neq v}$$

Figure 1: Tableau rules for STT

$$\begin{array}{c|c}
\multicolumn{2}{c}{pf,\ \neg p(\lambda x.\neg\neg fx)} \\
\multicolumn{2}{c}{[\mathcal{T}_{\mathrm{MAT}}]} \\
\multicolumn{2}{c}{f \neq (\lambda x.\neg\neg fx)} \\
\multicolumn{2}{c}{[\mathcal{T}_{\mathrm{FE}}]} \\
\multicolumn{2}{c}{fx \neq \neg\neg fx} \\
\multicolumn{2}{c}{[\mathcal{T}_{\mathrm{BE}}]} \\
\hline
fx,\ \neg\neg\neg fx & \neg fx,\ \neg\neg fx \\
[\mathcal{T}_{\neg\neg}] & [\mathcal{T}_{\neg\neg}] \\
\neg fx & fx \\
[\mathcal{T}_{\mathrm{MAT}}] & [\mathcal{T}_{\mathrm{MAT}}] \\
x \neq x & x \neq x \\
[\mathcal{T}_{\mathrm{DEC}}] & [\mathcal{T}_{\mathrm{DEC}}]
\end{array}$$

Figure 2: Tableau refuting $\{pf, \neg p(\lambda x.\neg\neg fx)\}$ where $p : (\alpha o)o$ and $f : \alpha o$

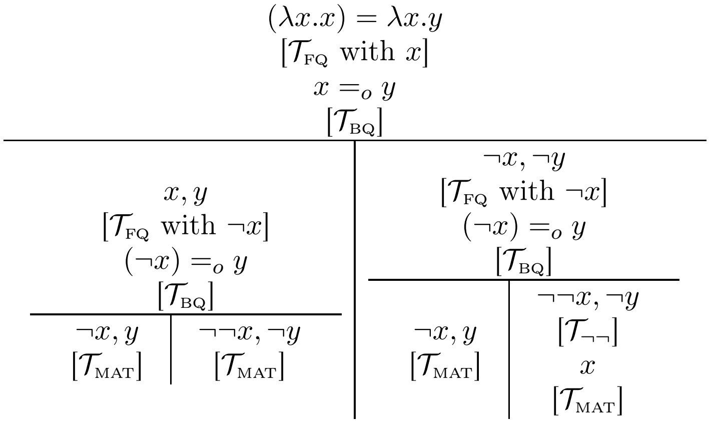
A remark on the names of the rules: $\mathcal{T}_{\mathrm{MAT}}$ is called the mating rule, $\mathcal{T}_{\mathrm{DEC}}$ the decomposition rule, $\mathcal{T}_{\mathrm{CON}}$ the confrontation rule, $\mathcal{T}_{\mathrm{BQ}}$ the Boolean equality rule, $\mathcal{T}_{\mathrm{BE}}$ the Boolean extensionality rule, $\mathcal{T}_{\mathrm{FQ}}$ the functional equality rule, and $\mathcal{T}_{\mathrm{FE}}$ the functional extensionality rule.

Proposition 5.1 (Soundness)**.** *Every refutable branch is unsatisfiable.*

Proof. Let $A/A_1 \dots A_n$ be an instance of a rule of $\mathcal{T}$ such that A is satisfiable. It suffices to show that one of the branches $A_1, \dots, A_n$ is satisfiable. Straightforward. □

We will show that the tableau calculus $\mathcal{T}$ is *complete*, that is, can refute every finite unsatisfiable branch. The rules of $\mathcal{T}$ are designed such that we obtain a strong completeness result. For practical purposes one can of course include rules that close branches including $s, \neg s$ or $s \neq s$.

To avoid redundancy, our definition of STT only covers the logical constants $\neg$ and $=_\sigma$. Adding further constants such as $\wedge$, $\vee$, $\rightarrow$, $\forall_\sigma$ and $\exists_\sigma$ is straightforward. In fact, all logical constants can be expressed with the identities $=_\sigma$ [1]. We have included $\neg$ since we need



$(\lambda x.x) = \lambda x.y$
$[\mathcal{T}_{\text{FQ}}$ with $x]$
$x =_o y$
$[\mathcal{T}_{\text{BQ}}]$
x, y
$[\mathcal{T}_{\text{FQ}}$ with $\neg x]$
$(\neg x) =_o y$
$[\mathcal{T}_{\text{BQ}}]$
$\neg x, y$
$[\mathcal{T}_{\text{MAT}}]$
$\neg\neg x, \neg y$
$[\mathcal{T}_{\text{MAT}}]$
$\neg x, \neg y$
$[\mathcal{T}_{\text{FQ}}$ with $\neg x]$
$(\neg x) =_o y$
$[\mathcal{T}_{\text{BQ}}]$
$\neg x, y$
$[\mathcal{T}_{\text{MAT}}]$
$\neg\neg x, \neg y$
$[\mathcal{T}_{\neg\neg}]$
x
$[\mathcal{T}_{\text{MAT}}]$

Figure 3: Tableau refuting $(\lambda x.x) = \lambda x.y$ where $x, y : o$

$\mathcal{E}_{\neg\neg}$ If $\neg\neg s$ is in E, then s is in E.

$\mathcal{E}_{\text{BQ}}$ If $s =_o t$ is in E, then either s and t are in E or $\neg s$ and $\neg t$ are in E.

$\mathcal{E}_{\text{BE}}$ If $s \neq_o t$ is in E, then either s and $\neg t$ are in E or $\neg s$ and t are in E.

$\mathcal{E}_{\text{FQ}}$ If $s =_{\sigma\tau} t$ is in E, then $[su] = [tu]$ is in E for every normal $u : \sigma$.

$\mathcal{E}_{\text{FE}}$ If $s \neq_{\sigma\tau} t$ is in E, then $[sx] \neq [tx]$ is in E for some variable x.

$\mathcal{E}_{\text{MAT}}$ If $xs_1 \dots s_n$ and $\neg xt_1 \dots t_n$ are in E, then $n \geq 1$ and $s_i \neq t_i$ is in E for some $i \in \{1, \dots, n\}$. Note that if $n = 0$, this means if $\neg x \in E$, then $x \notin E$.

$\mathcal{E}_{\text{DEC}}$ If $xs_1 \dots s_n \neq_\alpha xt_1 \dots t_n$ is in E, then $n \geq 1$ and $s_i \neq t_i$ is in E for some $i \in \{1, \dots, n\}$. Note that if $n = 0$, this means $x \neq_\alpha x \notin E$.

$\mathcal{E}_{\text{CON}}$ If $s =_\alpha t$ and $u \neq_\alpha v$ are in E,
then either $s \neq u$ and $t \neq u$ are in E or $s \neq v$ and $t \neq v$ are in E.

Figure 4: Evidence conditions

it for the formulation of the tableau calculus. The refutation in Figure 3 suggests that the elimination of $\neg$ is not straightforward.

6. Evidence

A branch E is *evident* if it satisfies the *evidence conditions* in Figure 4. The evidence conditions correspond to the tableau rules and are designed such that every branch that is closed under the tableau rules is either closed or evident. We will show that evident branches are satisfiable.

A branch E is *complete* if for every normal formula s either s or $\neg s$ is in E. The cut-freeness of $\mathcal{T}$ shows in the fact that there are many evident sets that are not complete. For instance, $\{pf,\ \neg p(\lambda x.\neg fx),\ f \neq \lambda x.\neg fx,\ fx \neq \neg fx,\ \neg fx\}$ is an incomplete evident branch if $p : (\sigma o)o$.

6.1. **Discriminants.** Given an evident branch E, we will construct a value system whose admissible logical interpretations are models of E. We start by defining the values for the sorts, which we call discriminants. Discriminants first appeared in [15].

Let E be a fixed evident branch in the following. A term $u \in \Lambda_\alpha$ is *α-discriminating in E* if there is some term t such that either $u \neq_\alpha t$ or $t \neq_\alpha u$ is in E. An *α-discriminant* is a maximal set a of discriminating terms of type α such that there is no disequation $s{\neq}t \in E$ such that $s, t \in a$. We write $s \sharp t$ if E contains the disequation $s{\neq}t$ or $t{\neq}s$.

In [12] a sort was interpreted using maximally compatible sets of terms of the sort (where s and t are compatible unless $s \sharp t$). The idea is that the set E insists that certain terms cannot be equal, but leaves open that other terms ultimately may be identified by the interpretation. In particular, two compatible terms s and t may be identified by taking a maximally compatible set of terms containing both s and t as a value. It is not difficult to see that a maximally compatible set is simply the union of an α-discriminant with all terms of sort α that are not α-discriminating. We now find that it is clearer to use α-discriminants as values instead of maximally compatible sets. In particular, it is easier to count the number of α-discriminants, as we now show.

Example 6.1. Suppose $E = \{x{\neq}y,\ x{\neq}z,\ y{\neq}z\}$ and $x, y, z : \alpha$. There are 3 α-discriminants: $\{x\}$, $\{y\}$, $\{z\}$.

Example 6.2. Suppose $E = \{\, a_n \neq_\alpha b_n \mid n \in \mathbb{N} \,\}$ where the a_n and b_n are pairwise distinct variables. Then E is evident and there are uncountably many α-discriminants.

Proposition 6.3. *If E contains exactly n disequations at α, then there are at most 2^n α-discriminants. If E contains no disequation at α, then $\emptyset$ is the only α-discriminant.*

Proposition 6.4. *Let a and b be different discriminants. Then:*

(1) *a and b are separated by a disequation in E, that is, there exist terms $s \in a$ and $t \in b$ such that $s \sharp t$.*
(2) *a and b are not connected by an equation in E, that is, there exist no terms $s \in a$ and $t \in b$ such that $(s{=}t) \in E$.*

Proof. The first claim follows by contradiction. Suppose there are no terms $s \in a$ and $t \in b$ such that $s \sharp t$. Let $s \in a$. Then $s \in b$ since b is a maximal set of discriminating terms. Thus $a \subseteq b$ and hence $a = b$ since a is maximal. Contradiction.

The second claim also follows by contradiction. Suppose there is an equation $(s_1{=}s_2) \in E$ such that $s_1 \in a$ and $s_2 \in b$. By the first claim we have terms $s \in a$ and $t \in b$ such that $s \sharp t$. By $\mathcal{E}_{\text{CON}}$ we have $s_1 \sharp s$ or $s_2 \sharp t$. Contradiction since a and b are discriminants. □

6.2. **Compatibility.** For our proofs we need an auxiliary notion for evident branches that we call compatibility. Let E be a fixed evident branch in the following. We define relations $\|_\sigma \subseteq \Lambda_\sigma \times \Lambda_\sigma$ by induction on types:

$$s \parallel_o t :\Longleftrightarrow \{[s], \neg[t]\} \not\subseteq E \text{ and } \{\neg[s], [t]\} \not\subseteq E$$

$$s \parallel_\alpha t :\Longleftrightarrow \text{not } [s] \sharp [t]$$

$$s \parallel_{\sigma\tau} t :\Longleftrightarrow su \parallel_\tau tv \text{ whenever } u \parallel_\sigma v$$

We say that s and t are *compatible* if $s \parallel t$.

Lemma 6.5 (Compatibility)**.**
For $n \geq 0$ and all terms s, t, $xs_1 \dots s_n$, $xt_1 \dots t_n$ of type σ:

(1) *We do not have both $s \parallel_\sigma t$ and $[s]\sharp[t]$.*
(2) *Either $xs_1 \dots s_n \parallel_\sigma xt_1 \dots t_n$ or $[s_i]\sharp[t_i]$ for some $i \in \{1, \dots, n\}$.*

Proof. By induction on σ. Case analysis.

$\sigma = o$. Claim (1) follows with $\mathcal{E}_{\text{BE}}$. Claim (2) follows with N3 and $\mathcal{E}_{\text{MAT}}$.

$\sigma = \alpha$. Claim (1) is trivial. Claim (2) follows with N3 and $\mathcal{E}_{\text{DEC}}$.

$\sigma = \tau\mu$. We show (1) by contradiction. Suppose $s \parallel_\sigma t$ and $[s]\sharp[t]$. By $\mathcal{E}_{\text{FE}}$ $[[s]x]\sharp[[t]x]$ for some variable x. By inductive hypothesis (2) we have $x \parallel_\tau x$. Hence $sx \parallel_\mu tx$. Contradiction by inductive hypothesis (1) and N2.

To show (2), suppose $xs_1 \dots s_n \nparallel_\sigma xt_1 \dots t_n$. Then there exist terms such that $u \parallel_\tau v$ and $xs_1 \dots s_n u \nparallel_\mu xt_1 \dots t_n v$. By inductive hypothesis (1) we know that $[u]\sharp[v]$ does not hold. Hence $[s_i]\sharp[t_i]$ for some $i \in \{1, \dots, n\}$ by inductive hypothesis (2). □

7. Model Existence

Let E be a fixed evident branch. We define a value system $\rhd$ for E:

$$s \rhd_o 0 :\Longleftrightarrow s \in \Lambda_o \text{ and } [s] \notin E$$

$$s \rhd_o 1 :\Longleftrightarrow s \in \Lambda_o \text{ and } \neg[s] \notin E$$

$$s \rhd_\alpha a :\Longleftrightarrow s \in \Lambda_\alpha,\ a \text{ is an } \alpha\text{-discriminant, and } [s] \in a \text{ if } [s] \text{ is discriminating}$$

Note that N1 ensures the property $s \rhd_\beta a$ iff $[s] \rhd_\beta a$.

Proposition 7.1. *For all variables x_o, either $x \rhd 0$ and $\neg x \rhd 1$ or $x \rhd 1$ and $\neg x \rhd 0$. In particular, $\mathcal{D}o = \{0, 1\}$.*

Proof. By $\mathcal{E}_{\text{MAT}}$ either $x \notin E$ or $\neg x \notin E$. If $x \notin E$, then $x \rhd 0$ and $\neg x \rhd 1$ by N3 and $\mathcal{E}_{\neg\neg}$. If $\neg x \notin E$, then $x \rhd 1$ and $\neg x \rhd 0$ by N3. □

Lemma 7.2. *A logical assignment is a model of E if it is admissible for $\rhd$.*

Proof. Let $\mathcal{I}$ be a logical assignment that is admissible for $\rhd$, and let $s \in E$. By Theorem 3.4 we know that $\mathcal{I}$ is an interpretation and that $s \rhd_o \hat{\mathcal{I}}s$. Thus $\hat{\mathcal{I}}s \neq 0$ since $s \in E$. Hence $\hat{\mathcal{I}}s = 1$. □

It remains to show that $\rhd$ admits logical interpretations. First we show that all sets $\mathcal{D}\sigma$ are nonempty. To do so, we prove that compatible equi-typed terms have a common value. A set T of equi-typed terms is *compatible* if $s \parallel t$ for all terms $s, t \in T$. We write $T \rhd_\sigma a$ if $T \subseteq \Lambda_\sigma$, $a \in \mathcal{D}\sigma$, and $t \rhd a$ for every $t \in T$.

Lemma 7.3 (Common Value)**.** *Let $T \subseteq \Lambda_\sigma$. Then T is compatible if and only if there exists a value a such that $T \rhd_\sigma a$.*

Proof. By induction on σ.

$\sigma = \alpha$, $\Rightarrow$. Let T be compatible. Then there exists an α-discriminant a that contains all the α-discriminating terms in $\{\, [t] \mid t \in T \,\}$. Clearly, $T \rhd a$.

$\sigma = \alpha$, $\Leftarrow$. Suppose $T \rhd a$ and T is not compatible. Then there are terms $s, t \in T$ such that $([s]{\neq}[t]) \in E$. Thus $[s]$ and $[t]$ cannot be both in a. This contradicts $s, t \in T \rhd a$ since $[s]$ and $[t]$ are discriminating.

$\sigma = o$, $\Rightarrow$. By contraposition. Suppose $T \ntriangleright 0$ and $T \ntriangleright 1$. Then there are terms $s, t \in T$ such that $[s], \neg[t] \in E$. Thus $s \nparallel t$. Hence T is not compatible.

$\sigma = o$, $\Leftarrow$. By contraposition. Suppose $s \nparallel_o t$ for $s, t \in T$. Then $[s], \neg[t] \in E$ without loss of generality. Hence $s \ntriangleright 0$ and $t \ntriangleright 1$. Thus $T \ntriangleright 0$ and $T \ntriangleright 1$.

$\sigma = \tau\mu$, $\Rightarrow$. Let T be compatible. We define $T_a := \{\, ts \mid t \in T,\ s \triangleright_\tau a \,\}$ for every value $a \in \mathcal{I}\tau$ and show that T_a is compatible. Let $t_1, t_2 \in T$ and $s_1, s_2 \triangleright_\tau a$. It suffices to show $t_1 s_1 \parallel t_2 s_2$. By the inductive hypothesis $s_1 \parallel_\tau s_2$. Since T is compatible, $t_1 \parallel t_2$. Hence $t_1 s_1 \parallel t_2 s_2$.

By the inductive hypothesis we now know that for every $a \in \mathcal{I}\tau$ there is a $b \in \mathcal{I}\mu$ such that $T_a \triangleright_\mu b$. Hence there is a function $f \in \mathcal{I}\sigma$ such that $T_a \triangleright_\mu fa$ for every $a \in \mathcal{I}\tau$. Thus $T \triangleright_\sigma f$.

$\sigma = \tau\mu$, $\Leftarrow$. Let $T \triangleright_\sigma f$ and $s, t \in T$. We show $s \parallel_\sigma t$. Let $u \parallel_\tau v$. It suffices to show $su \parallel_\mu tv$. By the inductive hypothesis $u, v \triangleright_\tau a$ for some value a. Hence $su, tv \triangleright_\mu fa$. Thus $su \parallel_\mu tv$ by the inductive hypothesis. □

Lemma 7.4 (Admissibility)**.** *For every variable $x : \sigma$ there is some $a \in \mathcal{D}\sigma$ such that $x \triangleright a$. In particular, $\mathcal{D}\sigma$ is a nonempty set for every type σ.*

Proof. Let $x : \sigma$ be a variable. By Lemma 6.5 (2) we know $x \parallel_\sigma x$. Hence $\{x\}$ is compatible. By Lemma 7.3 there exists a value a such that $x \triangleright_\sigma a$. The claim follows since $a \in \mathcal{D}\sigma$ by definition of $\mathcal{D}\sigma$. □

Lemma 7.5 (Functionality)**.** *If $s \triangleright_\sigma a$, $t \triangleright_\sigma b$, and $(s{=}t) \in E$, then $a = b$.*

Proof. By contradiction and induction on σ. Assume $s \triangleright_\sigma a$, $t \triangleright_\sigma b$, $(s{=}t) \in E$, and $a \neq b$. Case analysis.

$\sigma = o$. By $\mathcal{E}_{\text{BQ}}$ either $s, t \in E$ or $\neg s, \neg t \in E$. Hence a and b are either both 1 or both 0. Contradiction.

$\sigma = \alpha$. Since $a \neq b$, there must be discriminating terms of type α. Since $(s{=}t) \in E$, we know by N3 and $\mathcal{E}_{\text{CON}}$ that s and t are normal and discriminating. Hence $s \in a$ and $t \in b$. Contradiction by Proposition 6.4 (2).

$\sigma = \tau\mu$. Since $a \neq b$, there is some $c \in \mathcal{D}\tau$ such that $ac \neq bc$. By the definition of $\mathcal{D}\tau$ and Lemma 3.1 there is a normal term u such that $u \triangleright_\tau c$. Hence $su \triangleright ac$ and $tu \triangleright bc$. By Proposition 3.1 $[su] {\triangleright_\mu} ac$ and $[tu] {\triangleright_\mu} bc$. By $\mathcal{E}_{\text{FQ}}$ the equation $[su] = [tu]$ is in E. Contradiction by the inductive hypothesis. □

We now define the canonical interpretations for the logical constants:

$$\mathcal{L}(\neg) := \lambda a{\in}\mathcal{D}o.\ \text{if } a{=}1 \text{ then } 0 \text{ else } 1$$

$$\mathcal{L}(=_\sigma) := \lambda a{\in}\mathcal{D}\sigma.\ \lambda b{\in}\mathcal{D}\sigma.\ \text{if } a{=}b \text{ then } 1 \text{ else } 0$$

Lemma 7.6 (Logical Constants)**.** *$c \triangleright \mathcal{L}(c)$ for every logical constant c.*

Proof. We show $\neg {\triangleright} \mathcal{L}(\neg)$ by contradiction. Let $s {\triangleright_o} a$ and assume $\neg s \ntriangleright \mathcal{L}(\neg)a$. Case analysis.

- $a = 0$. Then $[s] \notin E$ and $\neg[\neg s] \in E$. Contradiction by N3 and $\mathcal{E}_{\neg\neg}$.
- $a = 1$. Then $\neg[s] \notin E$ and $[\neg s] \in E$. Contradiction by N3.

Finally, we show $(=_\sigma) \triangleright \mathcal{L}(=_\sigma)$ by contradiction. Let $s \triangleright_\sigma a$, $t \triangleright_\sigma b$, and $(s{=_\sigma}t) \ntriangleright \mathcal{L}(=_\sigma)ab$. Case analysis.

- $a = b$. Then $[s]\sharp[t]$ by N3 and $s, t \triangleright a$. Thus $s \parallel t$ by Lemma 7.3. Contradiction by Lemma 6.5 (1).

- $a \neq b$. Then $([s]{=}[t]) \in E$ by N3. Hence $a = b$ by Proposition 3.1 and Lemma 7.5. Contradiction. □

Theorem 7.7 (Model Existence)**.** *Every evident branch is satisfiable. Moreover, every complete evident branch has a surjective model, and every finite evident branch has a finite model.*

Proof. Let E be an evident branch and $\triangleright$ be the value system for E. By Proposition 7.1, Lemma 7.4, and Lemma 7.6 we have a logical interpretation $\mathcal{I}$ that is admissible for $\triangleright$. By Lemma 7.2 $\mathcal{I}$ is a model of E.

Let E be complete. By Theorem 3.4 we know that $\mathcal{I}$ is surjective if $\triangleright$ is functional. Let $s \triangleright_\beta a$ and $s \triangleright_\beta b$. We show $a = b$. By Proposition 3.1 we can assume that s is normal. Thus $s{=}s$ is normal by N3. Since $\mathcal{I}$ is a model of E, we know that the formula $s{\neq}s$ is not in E. Since E is complete, we know that $s{=}s$ is in E. By Lemma 7.5 we have $a = b$.

If E is finite, $\mathcal{I}\alpha = \mathcal{D}\alpha$ is finite by Proposition 6.3. □

8. Abstract Consistency

We now extend the model existence result for evident branches to abstract consistency classes, following the corresponding development for first-order logic [25]. Notions of abstract consistency for simple type theory have been previously considered in [2, 21, 22, 4, 9, 6, 7, 12, 13]. Equality was treated as Leibniz equality in [2]. Abstract consistency conditions for primitive equality corresponding to reflexivity and substutivity properties were given by Benzmüller in [4, 5]. A primitive identity predicate $=_\sigma$ was considered in [6] but the abstract consistency conditions for $=_\sigma$ essentially reduced it to Leibniz equality. Conditions for $=_\sigma$ analogous to $\mathcal{C}_{\text{CON}}$ first appeared in [12].

An *abstract consistency class* is a set Γ of branches such that every branch $A \in \Gamma$ satisfies the conditions in Figure 5. An abstract consistency class Γ is *complete* if for every branch $A \in \Gamma$ and every normal formula s either $A \cup \{s\}$ or $A \cup \{\neg s\}$ is in Γ. The completeness condition was called "saturation" in [6]. As discussed in [8] and the conclusion of [6], the condition corresponds to having a cut rule in a calculus. In [7] conditions analogous to $\mathcal{C}_{\text{DEC}}$ and $\mathcal{C}_{\text{MAT}}$ appear (using Leibniz equality) and a model existence theorem is proven with these conditions replacing saturation. The use of Leibniz equality means that there was still not a cut-free treatment of equality in [7].

Proposition 8.1. *Let A be a branch. Then A is evident if and only if $\{A\}$ is an abstract consistency class. Moreover, A is a complete evident branch if and only if $\{A\}$ is a complete abstract consistency class.*

Lemma 8.2 (Extension Lemma)**.** *Let Γ be an abstract consistency class and $A \in \Gamma$. Then there exists an evident branch E such that $A \subseteq E$. Moreover, if Γ is complete, a complete evident branch E exists such that $A \subseteq E$.*

Proof. Let $u_0, u_1, u_2, \ldots$ be an enumeration of all normal formulas. We construct a sequence $A_0 \subseteq A_1 \subseteq A_2 \subseteq \cdots$ of branches such that every $A_n \in \Gamma$. Let $A_0 := A$. We define A_{n+1} by cases. If there is no $B \in \Gamma$ such that $A_n \cup \{u_n\} \subseteq B$, then let $A_{n+1} := A_n$. Otherwise, choose some $B \in \Gamma$ such that $A_n \cup \{u_n\} \subseteq B$. We consider two subcases.

(1) If u_n is of the form $s \neq_{\sigma\tau} t$, then choose A_{n+1} to be $B \cup \{[sx] \neq [tx]\} \in \Gamma$ for some variable x. This is possible since Γ satisfies $\mathcal{C}_{\text{FE}}$.

$\mathcal{C}_{\neg\neg}$ If $\neg\neg s$ is in A, then $A \cup \{s\}$ is in Γ.

$\mathcal{C}_{\text{BQ}}$ If $s =_o t$ is in A, then either $A \cup \{s, t\}$ or $A \cup \{\neg s, \neg t\}$ is in Γ.

$\mathcal{C}_{\text{BE}}$ If $s \neq_o t$ is in A, then either $A \cup \{s, \neg t\}$ or $A \cup \{\neg s, t\}$ is in Γ.

$\mathcal{C}_{\text{FQ}}$ If $s =_{\sigma\tau} t$ is in A,
then $A \cup \{[su] \neq [tu]\}$ is in Γ for every normal $u : \sigma$.

$\mathcal{C}_{\text{FE}}$ If $s \neq_{\sigma\tau} t$ is in A, then $A \cup \{[sx] \neq [tx]\}$ is in Γ for some variable x.

$\mathcal{C}_{\text{MAT}}$ If $xs_1 \dots s_n$ is in A and $\neg xt_1 \dots t_n$ is in A,
then $n \geq 1$ and $A \cup \{s_i \neq t_i\}$ is in Γ for some $i \in \{1, \dots, n\}$.

$\mathcal{C}_{\text{DEC}}$ If $xs_1 \dots s_n \neq_\alpha xt_1 \dots t_n$ is in A, then $n \geq 1$ and $A \cup \{s_i \neq t_i\}$ is in Γ for some $i \in \{1, \dots, n\}$.

$\mathcal{C}_{\text{CON}}$ If $s =_\alpha t$ and $u \neq_\alpha v$ are in A,
then either $A \cup \{s \neq u, t \neq u\}$ or $A \cup \{s \neq v, t \neq v\}$ is in Γ.

Figure 5: Abstract consistency conditions (must hold for every $A \in \Gamma$)

(2) If u_n is not of this form, then let A_{n+1} be B.

Let $E := \bigcup_{n \in \mathbb{N}} A_n$. We show that E satisfies the evidence conditions.

$\mathcal{E}_{\neg\neg}$ Assume $\neg\neg s$ is in E. Let n be such that $u_n = s$. Let $r \geq n$ be such that $\neg\neg s$ is in A_r. By $\mathcal{C}_{\neg\neg}$, $A_r \cup \{s\} \in \Gamma$. Since $A_n \cup \{s\} \subseteq A_r \cup \{s\}$, we have $s \in A_{n+1} \subseteq E$.

$\mathcal{E}_{\text{MAT}}$ Assume $xs_1 \dots s_n$ and $\neg xt_1 \dots t_n$ are in E. For each $i \in \{1, \dots, n\}$, let m_i be such that u_{m_i} is $s_i \neq t_i$. Let $r \geq m_1, \dots, m_n$ be such that $xs_1 \dots s_n$ and $\neg xt_1 \dots t_n$ are in A_r. By $\mathcal{C}_{\text{MAT}}$ $n \geq 1$ and there is some $i \in \{1, \dots, n\}$ such that $A_r \cup \{s_i \neq t_i\} \in \Gamma$. Since $A_{m_i} \cup \{s_i \neq t_i\} \subseteq A_r \cup \{s_i \neq t_i\}$, we have $(s_i \neq t_i) \in A_{m_i+1} \subseteq E$.

$\mathcal{E}_{\text{DEC}}$ Similar to $\mathcal{E}_{\text{MAT}}$

$\mathcal{E}_{\text{CON}}$ Assume $s =_\alpha t$ and $u \neq_\alpha v$ are in E. Let n, m, j, k be such that u_n is $s \neq u$, u_m is $t \neq u$, u_j is $s \neq v$ and u_k is $t \neq v$. Let $r \geq n, m, j, k$ be such that $s =_\alpha t$ and $u \neq_\alpha v$ are in A_r. By $\mathcal{C}_{\text{CON}}$ either $A_r \cup \{s \neq u, t \neq u\}$ or $A_r \cup \{s \neq v, t \neq v\}$ is in Γ. Assume $A_r \cup \{s \neq u, t \neq u\}$ is in Γ. Since $A_n \cup \{s \neq u\} \subseteq A_r \cup \{s \neq u, t \neq u\}$, we have $s \neq u \in A_{n+1} \subseteq E$. Since $A_m \cup \{t \neq u\} \subseteq A_r \cup \{s \neq u, t \neq u\}$, we have $t \neq u \in A_{m+1} \subseteq E$. Next assume $A_r \cup \{s \neq v, t \neq v\}$ is in Γ. By a similar argument we know $s \neq v$ and $t \neq v$ must be in E.

$\mathcal{E}_{\text{BQ}}$ Assume $s =_o t$ is in E. Let n, m, j, k be such that $u_n = s$, $u_m = t$, $u_j = \neg s$ and $u_k = \neg t$. Let $r \geq n, m, j, k$ be such that $s =_o t$ is in A_r. By $\mathcal{C}_{\text{BQ}}$ either $A_r \cup \{s, t\}$ or $A_r \cup \{\neg s, \neg t\}$ is in Γ. Assume $A_r \cup \{s, t\}$ is in Γ. Since $A_n \cup \{s\} \subseteq A_r \cup \{s, t\}$, we have $s \in E$. Since $A_m \cup \{t\} \subseteq A_r \cup \{s, t\}$, we have $t \in E$. Next assume $A_r \cup \{\neg s, \neg t\}$ is in Γ. Since $A_j \cup \{\neg s\} \subseteq A_r \cup \{\neg s, \neg t\}$, we have $\neg s \in E$. Since $A_k \cup \{\neg t\} \subseteq A_r \cup \{\neg s, \neg t\}$, we have $\neg t \in E$.

$\mathcal{E}_{\text{BE}}$ Similar to $\mathcal{E}_{\text{BQ}}$

$\mathcal{E}_{\text{FQ}}$ Assume $s =_{\sigma\tau} t$ is in E and $u : \sigma$ is normal. Let n be such that u_n is $[su] =_\tau [tu]$. Let $r \geq n$ be such that $s =_{\sigma\tau} t$ is in A_r. By $\mathcal{C}_{\text{FQ}}$ we know $A_r \cup \{[su] =_\tau [tu]\}$ is in Γ. Hence $[su] =_\tau [tu]$ is in A_{n+1} and also in E.

$\mathcal{E}_{\text{FE}}$ Assume $s \neq_{\sigma\tau} t$ is in E. Let n be such that u_n is $s \neq_{\sigma\tau} t$. Let $r \geq n$ be such that $s \neq_{\sigma\tau} t$ is in A_r. Since $A_n \cup \{u_n\} \subseteq A_r$, there is some variable x such that $[sx] \neq_\tau [tx]$ is in $A_{n+1} \subseteq E$.

It remains to show that E is complete if Γ is complete. Let Γ be complete and s be a normal formula. We show that s or $\neg s$ is in E. Let m, n be such that $u_m = s$ and $u_n = \neg s$. We consider $m < n$. (The case $m > n$ is symmetric.) If $s \in A_n$, we have $s \in E$. If $s \notin A_n$, then $A_n \cup \{s\}$ is not in Γ. Hence $A_n \cup \{\neg s\}$ is in Γ since Γ is complete. Hence $\neg s \in A_{n+1} \subseteq E$. □

Theorem 8.3 (Model Existence)**.** *Every member of an abstract consistency class has a model, which is surjective if the consistency class is complete.*

Proof. Let $A \in \Gamma$ where Γ is an abstract consistency class. By Lemma 8.2 we have an evident set E such that $A \subseteq E$, where E is complete if Γ is complete. The claim follows with Theorem 7.7. □

9. Completeness

It is now straightforward to prove the completeness of the tableau calculus $\mathcal{T}$. Let $\Gamma_{\mathcal{T}}$ be the set of all finite branches that are not refutable.

Lemma 9.1. $\Gamma_{\mathcal{T}}$ *is an abstract consistency class.*

Proof. We have to show that $\Gamma_{\mathcal{T}}$ satisfies the abstract consistency conditions.

$\mathcal{C}_{\neg\neg}$ Assume $\neg\neg s$ is in A and $A \cup \{s\} \notin \Gamma_{\mathcal{T}}$. Then we can refute A using $\mathcal{T}_{\neg\neg}$.

$\mathcal{C}_{\text{MAT}}$ Assume $\{xs_1 \dots s_n, \neg xt_1 \dots t_n\} \subseteq A$ and $A \cup \{s_i \neq t_i\} \notin \Gamma_{\mathcal{T}}$ for all $i \in \{1, \dots, n\}$. Then we can refute A using $\mathcal{T}_{\text{MAT}}$.

$\mathcal{C}_{\text{DEC}}$ Assume $xs_1 \dots s_n \neq_\alpha xt_1 \dots t_n$ is in A and $A \cup \{s_i \neq t_i\} \notin \Gamma_{\mathcal{T}}$ for all $i \in \{1, \dots, n\}$. Then we can refute A using $\mathcal{T}_{\text{DEC}}$.

$\mathcal{C}_{\text{CON}}$ Assume $s =_\alpha t$ and $u \neq_\alpha v$ are in A but $A \cup \{s \neq u, t \neq u\}$ and $A \cup \{s \neq v, t \neq v\}$ are not in $\Gamma_{\mathcal{T}}$. Then we can refute A using $\mathcal{T}_{\text{CON}}$.

$\mathcal{C}_{\text{BQ}}$ Assume $s =_o t$ is in A, $A \cup \{s, t\} \notin \Gamma_{\mathcal{T}}$ and $A \cup \{\neg s, \neg t\} \notin \Gamma_{\mathcal{T}}$. Then we can refute A using $\mathcal{T}_{\text{BQ}}$.

$\mathcal{C}_{\text{BE}}$ Assume $s \neq_o t$ is in A, $A \cup \{s, \neg t\} \notin \Gamma_{\mathcal{T}}$ and $A \cup \{\neg s, t\} \notin \Gamma_{\mathcal{T}}$. Then we can refute A using $\mathcal{T}_{\text{BE}}$.

$\mathcal{C}_{\text{FQ}}$ Let $(s =_{\sigma\tau} t) \in A \in \Gamma_{\mathcal{T}}$. Suppose $A \cup \{[su]{=}[tu]\} \notin \Gamma_{\mathcal{T}}$ for some normal $u \in \Lambda_\sigma$. Then $A \cup \{[su]{=}[tu]\}$ is refutable and so A is refutable by $\mathcal{T}_{\text{FQ}}$.

$\mathcal{C}_{\text{FE}}$ Let $(s {\neq_{\sigma\tau}} t) \in A \in \Gamma_{\mathcal{T}}$. Suppose $A \cup \{[sx]{\neq}[tx]\} \notin \Gamma_{\mathcal{T}}$ for every variable $x : \sigma$. Then $A \cup \{[sx]{\neq}[tx]\}$ is refutable for every $x : \sigma$. Hence A is refutable using $\mathcal{T}_{\text{FE}}$ and the finiteness of A. Contradiction. □

Theorem 9.2 (Completeness)**.** *Every unsatisfiable finite branch is refutable.*

Proof. By contradiction. Let A be an unsatisfiable finite branch that is not refutable. Then $A \in \Gamma_{\mathcal{T}}$ and hence A is satisfiable by Lemma 9.1 and Theorem 8.3. Contradiction. □

10. Compactness and Countable Models

It is known [18, 1] that simple type theory is compact and has the countable-model property. We use the opportunity and show how these properties follow with the results we already have. It is only for the existence of countable models that we make use of complete evident sets and complete abstract consistency classes.

A branch A is *sufficiently pure* if for every type σ there are infinitely many variables of type σ that do not occur free in the formulas of A. Let Γ_{C} be the set of all sufficiently pure branches A such that every finite subset of A is satisfiable. We write $\subseteq_f$ for the finite subset relation.

Lemma 10.1. *Let $A \in \Gamma_{\mathrm{C}}$ and $B_1, \dots, B_n$ be finite branches such that $A \cup B_i \notin \Gamma_{\mathrm{C}}$ for all $i \in \{1, \dots, n\}$. Then there exists a finite branch $A' \subseteq_f A$ such that $A' \cup B_i$ is unsatisfiable for all $i \in \{1, \dots, n\}$.*

Proof. By the assumption, we have for every $i \in \{1, \dots, n\}$ a finite and unsatisfiable branch $C_i \subseteq A \cup B_i$. The branch $A' := (C_1 \cup \dots \cup C_n) \cap A$ satisfies the claim. □

Lemma 10.2. Γ_{C} *is a complete abstract consistency class.*

Proof. We verify the abstract consistency conditions using Lemma 10.1 tacitly.

$\mathcal{C}_{\neg\neg}$ Assume $\neg\neg s$ is in A and $A \cup \{s\} \notin \Gamma_{\mathrm{C}}$. There is some $A' \subseteq_{\mathrm{f}} A$ such that $A' \cup \{s\}$ is unsatisfiable. There is a model of $A' \cup \{\neg\neg s\} \subseteq_{\mathrm{f}} A$. This is also a model of $A' \cup \{s\}$, contradicting our choice of A'.

$\mathcal{C}_{\mathrm{MAT}}$ Assume $xs_1 \dots s_n$ and $\neg xt_1 \dots t_n$ are in A and $A \cup \{s_i \neq t_i\} \notin \Gamma_{\mathrm{C}}$ for all $i \in \{1, \dots, n\}$. There is some $A' \subseteq_{\mathrm{f}} A$ such that $A' \cup \{s_i \neq t_i\}$ is unsatisfiable for all $i \in \{1, \dots, n\}$. There is a model $\mathcal{I}$ of $A' \cup \{xs_1 \dots s_n, \neg xt_1 \dots t_n\} \subseteq_{\mathrm{f}} A$. Since $\hat{\mathcal{I}}(xs_1 \dots s_n) \neq \hat{\mathcal{I}}(xt_1 \dots t_n)$, we must have $\hat{\mathcal{I}}(s_i) \neq \hat{\mathcal{I}}(t_i)$ for some $i \in \{1, \dots, n\}$ (and in particular n must not be 0). Thus $\mathcal{I}$ models $A' \cup \{s_i \neq t_i\}$, contradicting our choice of A'.

$\mathcal{C}_{\mathrm{DEC}}$ Similar to $\mathcal{C}_{\mathrm{MAT}}$

$\mathcal{C}_{\mathrm{CON}}$ Assume $s =_\alpha t$ and $u \neq_\alpha v$ are in A, $A \cup \{s \neq u, t \neq u\} \notin \Gamma_{\mathrm{C}}$ and $A \cup \{s \neq v, t \neq v\} \notin \Gamma_{\mathrm{C}}$. There is some $A' \subseteq_{\mathrm{f}} A$ such that $A' \cup \{s \neq u, t \neq u\}$ and $A' \cup \{s \neq v, t \neq v\}$ are unsatisfiable. There is a model $\mathcal{I}$ of $A' \cup \{s = t, u \neq v\} \subseteq_{\mathrm{f}} A$. Since $\hat{\mathcal{I}}(s) = \hat{\mathcal{I}}(t)$ and $\hat{\mathcal{I}}(u) \neq \hat{\mathcal{I}}(v)$, we either have $\hat{\mathcal{I}}(s) \neq \hat{\mathcal{I}}(u)$ and $\hat{\mathcal{I}}(t) \neq \hat{\mathcal{I}}(u)$ or $\hat{\mathcal{I}}(s) \neq \hat{\mathcal{I}}(v)$ and $\hat{\mathcal{I}}(t) \neq \hat{\mathcal{I}}(v)$. Hence $\mathcal{I}$ models either $A' \cup \{s \neq u, t \neq u\}$ or $A' \cup \{s \neq v, t \neq v\}$, contradicting our choice of A'.

$\mathcal{C}_{\mathrm{BQ}}$ Assume $s =_o t$ is in A, $A \cup \{s, t\} \notin \Gamma_{\mathrm{C}}$ and $A \cup \{\neg s, \neg t\} \notin \Gamma_{\mathrm{C}}$. There is some $A' \subseteq_{\mathrm{f}} A$ such that $A' \cup \{s, t\}$ and $A' \cup \{\neg s, \neg t\}$ are unsatisfiable. There is a model of $A' \cup \{s =_o t\} \subseteq_{\mathrm{f}} A$. This is also a model of $A' \cup \{s, t\}$ or $A' \cup \{\neg s, \neg t\}$.

$\mathcal{C}_{\mathrm{BE}}$ Assume $s \neq_o t$ is in A, $A \cup \{s, \neg t\} \notin \Gamma_{\mathrm{C}}$ and $A \cup \{\neg s, t\} \notin \Gamma_{\mathrm{C}}$. There is some $A' \subseteq_{\mathrm{f}} A$ such that $A' \cup \{s, \neg t\}$ and $A' \cup \{\neg s, t\}$ are unsatisfiable. There is a model of $A' \cup \{s \neq_o t\} \subseteq_{\mathrm{f}} A$. This is also a model of $A' \cup \{s, \neg t\}$ or $A' \cup \{\neg s, t\}$.

$\mathcal{C}_{\mathrm{FQ}}$ Assume $s =_{\sigma\tau} t$ is in A but $A \cup \{[su] =_\tau [tu]\}$ is not in Γ_{C} for some normal $u \in \Lambda_\sigma$. There is some $A' \subseteq_{\mathrm{f}} A$ such that $A' \cup \{[su] = [tu]\}$ is unsatisfiable. There is a model $\mathcal{I}$ of $A' \cup \{s = t\} \subseteq_{\mathrm{f}} A$. Since $\hat{\mathcal{I}}(s) = \hat{\mathcal{I}}(t)$, we know $\hat{\mathcal{I}}([su]) = \hat{\mathcal{I}}(su) = \hat{\mathcal{I}}(s)\hat{\mathcal{I}}(u) = \hat{\mathcal{I}}(t)\hat{\mathcal{I}}(u) = \hat{\mathcal{I}}(tu) = \hat{\mathcal{I}}([tu])$ using N4. Hence $\mathcal{I}$ is a model of $A' \cup \{[su] = [tu]\}$, a contradiction.

$\mathcal{C}_{\text{FE}}$ Assume $s \neq_{\sigma\tau} t$ is in A. Since A is sufficiently pure, there is a variable $x : \sigma$ which does not occur in A. Assume $A \cup \{[sx] \neq [tx]\} \notin \Gamma_{\text{C}}$. There is some $A' \subseteq_{\text{f}} A$ such that $A' \cup \{[sx] \neq [tx]\}$ is unsatisfiable. There is a model $\mathcal{I}$ of $A' \cup \{s \neq t\} \subseteq_{\text{f}} A$. Since $\hat{\mathcal{I}}(s) \neq \hat{\mathcal{I}}(t)$, there must be some $a \in \mathcal{I}\sigma$ such that $\hat{\mathcal{I}}(s)a \neq \hat{\mathcal{I}}(t)a$. Since x does not occur free in A, we know $\widehat{\mathcal{I}^x_a}(sx) \neq \widehat{\mathcal{I}^x_a}(tx)$ and $\mathcal{I}^x_a$ is a model of A'. Since $\widehat{\mathcal{I}^x_a}([sx]) = \widehat{\mathcal{I}^x_a}(sx)$ and $\widehat{\mathcal{I}^x_a}([tx]) = \widehat{\mathcal{I}^x_a}(tx)$ by N4, we conclude $\mathcal{I}^x_a$ is a model of $A' \cup \{[sx] \neq [tx]\}$, contradicting our choice of A'.

We show the completeness of Γ_{C} by contradiction. Let $A \in \Gamma_{\text{C}}$ and s be a normal formula such that $A \cup \{s\}$ and $A \cup \{\neg s\}$ are not in Γ_{C}. Then there exists $A' \subseteq_{\text{f}} A$ such that $A' \cup \{s\}$ and $A' \cup \{\neg s\}$ are unsatisfiable. Contradiction since A' is satisfiable. □

Theorem 10.3. *Let A be a branch such that every finite subset of A is satisfiable. Then A has a countable model.*

Proof. Without loss of generality we assume A is sufficiently pure. Then $A \in \Gamma_{\text{C}}$. Hence A has a countable model by Lemma 10.2 and Theorem 8.3. □

11. EFO Fragment

We now turn to the EFO fragment of STT as first reported in [14]. The EFO fragment contains first-order logic and enjoys the usual properties of first-order logic. We will show completeness and compactness with respect to standard models. We will also prove that countable models for evident EFO sets exist.

Suppose STT were given with $\neg$, $\rightarrow$, $=_\sigma$ and $\forall_\sigma$. Then the natural definition of EFO would restrict $=_\sigma$ and $\forall_\sigma$ to the case where σ is a base type. To avoid redundancy our definition of EFO will also exclude the case where $\sigma = o$.

Our definition of EFO assumes the logical constants $\neg : oo$, $\rightarrow : ooo$, $=_\alpha : \alpha\alpha o$ and $\forall_\alpha : (\alpha o)o$ where α ranges over sorts. We call these constants *EFO constants*. For an assignment to be logical we require that it interprets the logical constants as usual. In particular, $\mathcal{I}(\forall_\alpha)$ must be the function returning 1 iff its argument is the constant 1 function.

We say a term is *EFO* if it only contains the logical constants $\neg$, $\rightarrow$, $=_\alpha$ and $\forall_\alpha$. Let *EFO*$_\sigma$ be the set of EFO terms of type σ. A term is *quasi-EFO* if it is EFO or of the form $s \neq_\sigma t$ where s, t are EFO and σ is a type. A branch E is *EFO* if every member of E is quasi-EFO. The example tableau shown in Figure 2 only contains EFO branches.

The tableau rules in Figure 6 define a tableau calculus $\mathcal{F}$ for EFO branches up to restrictions on applicability given in Section 14. After showing a model existence theorem, we will precisely define the tableau calculus $\mathcal{F}$ and prove it is complete for EFO branches. The completeness result will be with respect to standard models. For some fragments of EFO the tableau calculus $\mathcal{F}$ will terminate, yielding decidability results.

12. EFO Evidence and Compatibility

We say an EFO branch E is evident if it satisfies the evidence conditions in Figure 4 and the following additional conditions.

$$\mathcal{F}_{\neg\neg}\ \frac{\neg\neg s}{s} \qquad \mathcal{F}_{\text{BE}}\ \frac{s \neq_o t}{s\,,\,\neg t \ \mid\ \neg s\,,\,t} \qquad \mathcal{F}_{\rightarrow}\ \frac{s \rightarrow t}{\neg s \mid t} \qquad \mathcal{F}_{\neg\rightarrow}\ \frac{\neg(s \rightarrow t)}{s\,,\,\neg t}$$

$$\mathcal{F}_{\text{MAT}}\ \frac{xs_1 \ldots s_n\,,\,\neg xt_1 \ldots t_n}{s_1 \neq t_1 \mid \cdots \mid s_n \neq t_n}\ n \geq 0 \qquad \mathcal{F}_{\text{DEC}}\ \frac{xs_1 \ldots s_n \neq_\alpha xt_1 \ldots t_n}{s_1 \neq t_1 \mid \cdots \mid s_n \neq t_n}\ n \geq 0$$

$$\mathcal{F}_{\text{FE}}\ \frac{s \neq_{\sigma\tau} t}{[sx] \neq [tx]}\ x : \sigma \text{ FRESH} \qquad \mathcal{F}_{\text{CON}}\ \frac{s =_\alpha t\,,\,u \neq_\alpha v}{s \neq u\,,\,t \neq u \mid s \neq v\,,\,t \neq v}$$

$$\mathcal{F}_{\forall}\ \frac{\forall_\alpha s}{[su]}\ u \in \text{EFO}_\alpha \text{ NORMAL} \qquad \mathcal{F}_{\neg\forall}\ \frac{\neg\forall_\alpha s}{\neg[sx]}\ x : \alpha \text{ FRESH}$$

Figure 6: Tableau rules for EFO

$\mathcal{E}_{\rightarrow}$ If $s \rightarrow t$ is in E, then $\neg s$ or t is in E.

$\mathcal{E}_{\neg\rightarrow}$ If $\neg(s \rightarrow t)$ is in E, then s and $\neg t$ are in E.

$\mathcal{E}_{\forall}$ If $\forall_\alpha s$ is in E, then $[su]$ is in E for every α-discriminating u in E.

$\mathcal{E}_{\forall}^{\emptyset}$ If $\forall_\alpha s$ is in E, then $[su]$ is in E for some normal EFO term $u : \alpha$.

$\mathcal{E}_{\neg\forall}$ If $\neg\forall_\alpha s$ is in E, then $\neg[sx]$ is in E for some variable x.

We say an EFO branch E is *EFO-complete* if for all normal $s \in \text{EFO}_o$ either $s \in E$ or $\neg s \in E$.

The condition $\mathcal{E}_{\forall}$ is the usual condition for universal quantifiers with instantiations restricted to α-discriminating terms. Since there may be no α-discriminating terms in E, we also include the condition $\mathcal{E}_{\forall}^{\emptyset}$ to ensure that at least one instantiation has been made. Without the condition $\mathcal{E}_{\forall}^{\emptyset}$, the set $\{\forall_\alpha x.\neg(y \rightarrow y)\}$ would be evident.

Let E be an evident EFO branch. Compatibility can be defined exactly as in Section 6.2 and Lemma 6.5 holds. In the proof of Lemma 13.8 below, we will need to know that if E has some α-discriminating term, then all α-discriminants are nonempty. Since α-discriminants are maximal sets of α-discriminating terms, it is enough to prove every α-discriminating term is compatible with itself. To be concrete, we must prove $s \neq_\alpha s$ is never in E. One way we could ensure this is to include it as an evidence condition and have a corresponding tableau rule of the form:

$$\mathcal{F}_{\neq}\ \frac{s \neq_\alpha s}{}$$

This was the choice taken in [14]. One drawback to including the rule $\mathcal{F}_{\neq}$ in the ground calculus is that a lifting lemma will be more difficult to show when one passes to a calculus with variables.

Another alternative is to remove the restriction on instantiations in the rule $\mathcal{F}_{\forall}$. If we do not restrict $\mathcal{F}_{\forall}$ to discriminating terms, then we can show the existence of a model without knowing a priori that α-discriminants are nonempty in the presence of α-discriminating terms.

In order to obtain a strong completeness result, we will not follow either of these alternatives. Instead we prove that all terms are compatible with themselves. First we prove EFO constants are compatible with themselves.

Lemma 12.1. *For every EFO constant c, $c \parallel c$.*

Proof. Case analysis. $\neg \parallel \neg$ follows from N3 and $\mathcal{E}_{\neg\neg}$. $\rightarrow\parallel\rightarrow$ follows from N3, $\mathcal{E}_{\rightarrow}$ and $\mathcal{E}_{\neg\rightarrow}$. $=_\alpha\parallel=_\alpha$ follows from N3 and $\mathcal{E}_{\text{CON}}$. We show $\forall_\alpha \parallel \forall_\alpha$. Let $s \parallel_{\alpha o} t$ be given. Assume $\forall s \nparallel \forall t$. Without loss of generality, assume $[\forall s]$ and $\neg[\forall t]$ are in E. By $\mathcal{E}_{\neg\forall}$ we have $\neg[tx]$ in E for some variable $x : \alpha$. By $\mathcal{E}_{\forall}^{\emptyset}$ we have $[su]$ in E for some normal EFO term u. Since $su \nparallel_o tx$, we must have $u \nparallel_\alpha x$. In particular, x must be an α-discriminating term. By $\mathcal{E}_{\forall}$ we have $[sx]$ is in E. Hence we must have $x \nparallel_\alpha x$, contradicting Lemma 6.5 (2). □

Next we prove compatibility respects normalization.

Lemma 12.2. *For all $s, t : \sigma$, $s \parallel_\sigma t$ iff $[s] \parallel_\sigma [t]$.*

Proof. Induction on types. At base types this follows from N1 and the definition of compatibility. Assume σ is $\tau\mu$. Let $u \parallel_\tau v$. By N2 and the inductive hypothesis (twice) we have $su \parallel tv$ iff $[su] \parallel [tv]$ iff $[[s]u] \parallel [[t]v]$ iff $[s]u \parallel [t]v$. Hence $s \parallel t$ iff $[s] \parallel [t]$. □

For two substitutions θ and ϕ we write $\theta \parallel \phi$ when $\text{Dom}\,\theta = \text{Dom}\,\phi$, $\theta x \parallel \phi x$ for every variable $x \in \text{Dom}\,\theta$ and $\theta c \parallel \phi c$ for every EFO constant $c \in \text{Dom}\,\theta$.

Lemma 12.3. *For all $s \in EFO_\sigma$, if $\theta \parallel \phi$, then $\hat{\theta}s \parallel \hat{\phi}s$.*

Proof. By induction on s. Case analysis.

s is a variable or an EFO constant in $\text{Dom}\,\theta$. The claim follows from $\theta \parallel \phi$ and S1.

s is a variable not in $\text{Dom}\,\theta$. The claim follows from S1 and Lemma 6.5 (2).

s is an EFO constant not in $\text{Dom}\,\theta$. The claim follows from S1 and Lemma 12.1.

$s = tu$. By inductive hypothesis $\hat{\theta}t \parallel \hat{\phi}t$ and $\hat{\theta}u \parallel \hat{\phi}u$. Hence $\hat{\theta}(tu) \parallel \hat{\phi}(tu)$ using S2.

$s = \lambda x.t$ where $x : \sigma$. Let $u \parallel v$ be given. We will prove $(\hat{\theta}s)u \parallel (\hat{\phi}s)v$. Using Lemma 12.2 and S3 it is enough to prove $\widehat{\theta^x_u}t \parallel \widehat{\phi^x_v}t$. This is the inductive hypothesis with θ^x_u and ϕ^x_v. □

Lemma 12.4. *For all $s \in EFO_\sigma$, $s \parallel s$.*

Proof. By Lemma 12.3 we have $\hat{\emptyset}s \parallel \hat{\emptyset}s$. We conclude $s \parallel s$ using Lemma 12.2 and S4. □

We can now prove α-discriminants are nonempty if E has some α-discriminating term.

Lemma 12.5. *If a is an α-discriminant and E has an α-discriminating term, then a is nonempty.*

Proof. Let s be α-discriminating. We know $s \parallel s$ by Lemma 12.4 and so $\{s\}$ is compatible. If a is empty, then $a \cup \{s\}$ is compatible, contradicting maximality of a. □

13. EFO Model Construction

Let E be an evident EFO branch. We inductively define a standard frame $\mathcal{D}$.

$$\begin{aligned} \mathcal{D}o &= \{0,1\} \\ \mathcal{D}\alpha &= \{a|a \text{ is an } \alpha\text{-discriminant}\} \\ \mathcal{D}(\sigma\tau) &= \mathcal{D}\sigma \to \mathcal{D}\tau \end{aligned}$$

We define a value system $\triangleright$ as for STT, but extend it to higher types using full function spaces.

$$\begin{aligned} & s \triangleright_o 0 :\Longleftrightarrow s \in \Lambda_o \text{ and } [s] \notin E \\ & s \triangleright_o 1 :\Longleftrightarrow s \in \Lambda_o \text{ and } \neg[s] \notin E \\ & s \triangleright_\alpha a :\Longleftrightarrow s \in \Lambda_\alpha,\ a \text{ is an } \alpha\text{-discriminant, and } [s] \in a \text{ if } [s] \text{ is discriminating} \\ & \quad \triangleright_{\sigma\tau} := \{\, (s,f) \in \Lambda_{\sigma\tau} \times (\mathcal{D}\sigma \to \mathcal{D}\tau) \mid \forall (t,a) \in \triangleright_\sigma \colon (st, fa) \in \triangleright_\tau \,\} \end{aligned}$$

In spite of the slightly different construction, many of the previous results still hold with essentially the same proofs as before.

Proposition 13.1. *$s \triangleright_\sigma a$ iff $[s] \triangleright_\sigma a$.*

Proof. Similar to Proposition 3.1. □

Lemma 13.2. *Let $\mathcal{I}$ be an assignment into $\mathcal{D}$ such that $x \triangleright \mathcal{I}x$ for all names x and θ be a substitution such that $\theta x \triangleright \mathcal{I}x$ for all $x \in \operatorname{Dom}\theta$. Then $s \in \operatorname{Dom}\hat{\mathcal{I}}$ and $\hat{\theta}s \triangleright \hat{\mathcal{I}}s$ for every term s.*

Proof. Similar to Lemma 3.3 □

Theorem 13.3. *Let $\mathcal{I}$ be an assignment into $\mathcal{D}$ such that $x \triangleright \mathcal{I}x$ for all names x. Then $\mathcal{I}$ is an interpretation such that $s \triangleright \hat{\mathcal{I}}s$ for all terms s.*

Proof. Follows from Proposition 13.1, Lemma 13.2 and property S4. □

Lemma 13.4. *A logical assignment $\mathcal{I}$ is a model of E if $x \triangleright \mathcal{I}x$ for every name x.*

Proof. Similar to Lemma 7.2 using Theorem 13.3. □

Lemma 13.5 (Common Value)**.** *Let $T \subseteq \Lambda_\sigma$. Then T is compatible if and only if there exists a value a such that $T \triangleright_\sigma a$.*

Proof. Similar to Lemma 7.3. □

Lemma 13.6 (Admissibility)**.** *For every variable $x : \sigma$ there is some $a \in \mathcal{D}\sigma$ such that $x \triangleright a$.*

Proof. Similar to Lemma 7.4 using Lemma 6.5 and Lemma 13.5. □

Lemma 13.7 (Functionality)**.** *If $s \triangleright_\alpha a$, $t \triangleright_\alpha b$, and $(s{=}t) \in E$, then $a = b$.*

Proof. Similar to Lemma 7.5 restricted only to sorts. □

As before $\mathcal{L}(c)$ is the canonical interpretation for each logical constant c. We now have the additional logical constants $\to$ and $\forall_\alpha$:

$$\mathcal{L}(\to) := \lambda a{\in}\mathcal{D}o.\ \lambda b{\in}\mathcal{D}o.\ \text{if } a{=}1 \text{ then } b \text{ else } 1$$
$$\mathcal{L}(\forall_\alpha) := \lambda f{\in}\mathcal{D}\alpha \to \mathcal{D}o.\ \text{if } f = (\lambda x \in \mathcal{D}\alpha.\ 1) \text{ then } 1 \text{ else } 0$$

Lemma 13.8 (Logical Constants). *$c \triangleright \mathcal{L}(c)$ for every logical constant c.*

Proof. Similar to Lemma 7.6. The proof for $\neg$ is the same. The proof for $\to$ uses N3, $\mathcal{E}_{\to}$ and $\mathcal{E}_{\neg\to}$. The proof for $=_\sigma$ requires a slight modification. Assume $s \triangleright_\sigma a$, $t \triangleright_\sigma b$, and $(s{=}_\sigma t) \not\triangleright \mathcal{L}(=_\sigma)ab$. Case analysis.

- $a = b$. Use Lemmas 13.5 and 6.5 (1).
- $a \neq b$. Then $([s]{=}[t]) \in E$ and so σ must be a sort α since E is EFO. This contradicts Lemma 13.7.

Finally, we prove $\forall_\alpha \triangleright \mathcal{L}(\forall_\alpha)$. Case analysis. Assume $s \triangleright_{\alpha o} f$ and $\forall_\alpha s \not\triangleright_o \mathcal{L}(\forall_\alpha)f$.

- $\mathcal{L}(\forall_\alpha)f = 1$. Then $\neg[\forall_\alpha s] \in E$ and so by N3, $\mathcal{E}_{\neg\forall}$ and N2 we have $\neg[sx] \in E$ for some variable $x : \alpha$. We know $\{x\}$ is compatible by Lemma 6.5 (2) and so by Lemma 13.5 there is some $a \in \mathcal{D}\alpha$ such that $x \triangleright a$. Thus $sx \triangleright 1$, contradicting $\neg[sx] \in E$.
- $\mathcal{L}(\forall_\alpha)f = 0$. Then $[\forall_\alpha s] \in E$ and there is some $a \in \mathcal{D}\alpha$ such that $fa = 0$. Suppose there are no α-discriminating terms. In this case a is empty and $u \triangleright a$ for any $u \in \Lambda_\alpha$. By N3, $\mathcal{E}^{\emptyset}_{\forall}$ and N2 we have $[su] \in E$ for some normal EFO term u. Hence $su \not\triangleright 0$, contradicting $s \triangleright f$ and $u \triangleright a$. Next suppose there are α-discriminating terms. In this case there is some $u \in a$ by Lemma 12.5. By N3, $\mathcal{E}_{\forall}$ and N2 we know $[su] \in E$. In this case we also have $su \not\triangleright 0$, again contradicting $s \triangleright f$ and $u \triangleright a$. □

Theorem 13.9 (EFO Model Existence). *Every evident EFO branch has a standard model. Every EFO-complete evident EFO branch has a standard model where each $\mathcal{D}\alpha$ is countable. Every finite evident EFO branch has a finite standard model.*

Proof. We use the frame $\mathcal{D}$ and relation $\triangleright$ defined above. We give an assignment $\mathcal{I}$ into $\mathcal{D}$. For each variable x we can choose $\mathcal{I}x$ such that $x \triangleright \mathcal{I}x$ using Lemma 13.6. For each logical constant c we choose $\mathcal{I}c = \mathcal{L}(c)$. By Lemma 13.8 we know $c \triangleright \mathcal{I}c$. $\mathcal{I}$ is a model of E by Lemma 13.4.

Suppose E is EFO-complete. We prove there are only countably many α-discriminants as follows. If there are no α-discriminating terms, then $\emptyset$ is the only α-discriminant. Otherwise, every α-discriminant is nonempty by Lemma 12.5. For each α-discriminant a, choose some $s_a \in a$. We prove the function mapping a to s_a is injective. Assume $a, b \in \mathcal{D}\alpha$ and $a \neq b$. By EFO-completeness of E and Proposition 6.4 we must have $s_a \neq s_b \in E$. If s_a and s_b were the same term, then E would be unsatisfiable. Hence s_a and s_b are different terms.

Finally, if E is finite, then for each sort α there will be only finitely many α-discriminants (by Proposition 6.3) and hence $\mathcal{D}\sigma$ will be finite for all σ. □

14. EFO Completeness

Let $\mathcal{F}$ be the tableau calculus given by taking all the rules from Figure 6 subject to the following restrictions.

- If $(s{\neq}t)$ is on a branch A, then $\mathcal{F}_{\text{FE}}$ can only be applied if there is no variable x such that $([sx] \neq [tx]) \in A$.

- If $\neg\forall_\alpha s$ is on a branch A, then $\mathcal{F}_{\neg\forall}$ can only be applied if there is no variable $x : \alpha$ such that $\neg[sx] \in A$.
- If $\forall_\alpha s$ is on a branch A and there are α-discriminating terms in A, then $\mathcal{F}_\forall$ can only be applied with these α-discriminating terms.
- If $\forall_\alpha s$ is on a branch A, $[su] \notin A$ for all normal $u \in \Lambda_\alpha$, some variable $x : \alpha$ occurs free in A and there are no α-discriminating terms in A, then $\mathcal{F}_\forall$ can only be applied with a variable $x : \alpha$ occurring free in A.
- If $\forall_\alpha s$ is on a branch A, $[su] \notin A$ for all normal $u \in \Lambda_\alpha$, no variable $x : \alpha$ occurs free in A and there are no α-discriminating terms in A, then $\mathcal{F}_\forall$ can only be applied with a variable $x : \alpha$.

The idea behind the restrictions on $\mathcal{F}_\forall$ is that only α-discriminating terms should be used as instantiations, except when there are no α-discriminating terms. In case there are no α-discriminating terms, at most one new variable $x : \alpha$ will be used as an instantiation term for each sort α. These restrictions will ensure that $\mathcal{F}$ terminates when given branches in certain fragments of EFO.

From now on we use the term *refutable* to refer to refutability in the calculus $\mathcal{F}$. That is, the set of *refutable branches* is the least set such that if $A/A_1 \dots A_n$ is an instance of a rule of $\mathcal{F}$ and $A_1, \dots, A_n$ are refutable, then A is refutable. The proof of soundness of $\mathcal{T}$ (see Proposition 5.1) extends to show soundness of $\mathcal{F}$.

Proposition 14.1 (Soundness of $\mathcal{F}$)**.** *Every refutable branch is unsatisfiable.*

An EFO abstract consistency class is a set Γ of EFO branches such that every branch $A \in \Gamma$ satisfies the conditions in Figure 5 and also the following conditions:

$\mathcal{C}_\rightarrow$ If $s \rightarrow t$ is in A, then $A \cup \{\neg s\}$ or $A \cup \{t\}$ is in Γ.

$\mathcal{C}_{\neg\rightarrow}$ If $\neg(s \rightarrow t)$ is in A, then $A \cup \{s, \neg t\}$ is in Γ.

$\mathcal{C}_\forall$ If $\forall_\alpha s$ is in A, then $A \cup \{[su]\}$ is in Γ for every α-discriminating u in A.

$\mathcal{C}_\forall^\emptyset$ If $\forall_\alpha s$ is in A, then $A \cup \{[su]\}$ is in Γ for some normal EFO term $u \in \Lambda_\alpha$.

$\mathcal{C}_{\neg\forall}$ If $\neg\forall_\alpha s$ is in A, then $A \cup \{\neg[sx]\}$ is in Γ for some variable x.

We say an abstract consistency class Γ is *EFO-complete* if for all $A \in \Gamma$ and all normal $s \in \mathrm{EFO}_o$ either $A \cup \{s\} \in \Gamma$ or $A \cup \{\neg s\} \in \Gamma$.

Let $\Gamma_{\mathcal{F}}^{EFO}$ be the set of all finite EFO branches that are not refutable.

Lemma 14.2. *$\Gamma_{\mathcal{F}}^{EFO}$ is an abstract consistency class.*

Proof. Similar to Lemma 9.1. We only check the new conditions: $\mathcal{C}_\rightarrow$, $\mathcal{C}_{\neg\rightarrow}$, $\mathcal{C}_\forall$, $\mathcal{C}_\forall^\emptyset$ and $\mathcal{C}_{\neg\forall}$.

$\mathcal{C}_\rightarrow$ Let $s \rightarrow t \in A \in \Gamma_{\mathcal{F}}^{\mathrm{EFO}}$. Suppose $A \cup \{\neg s\} \notin \Gamma_{\mathcal{F}}^{\mathrm{EFO}}$ and $A \cup \{t\} \notin \Gamma_{\mathcal{F}}^{\mathrm{EFO}}$. By $\mathcal{F}_\rightarrow$ we have A is refutable. Contradiction.

$\mathcal{C}_{\neg\rightarrow}$ If $\neg(s \rightarrow t) \in A$ and $A \cup \{s, \neg t\} \notin \Gamma_{\mathcal{F}}^{\mathrm{EFO}}$, then $A \notin \Gamma_{\mathcal{F}}^{\mathrm{EFO}}$ using the rule $\mathcal{F}_{\neg\rightarrow}$.

$\mathcal{C}_\forall$ Let $\forall_\alpha s \in A \in \Gamma_{\mathcal{F}}^{\mathrm{EFO}}$. Suppose $A \cup \{[su]\} \notin \Gamma_{\mathcal{T}}$ for some normal α-discriminating u. Then $A \cup \{[su]\}$ is refutable. Hence A can be refuted using $\mathcal{F}_\forall$ (with the restriction).

$\mathcal{C}_\forall^\emptyset$ Let $\forall_\alpha s \in A \in \Gamma_{\mathcal{F}}^{\mathrm{EFO}}$. If there is some α-discriminating term, then $\mathcal{C}_\forall^\emptyset$ follows from $\mathcal{C}_\forall$. Assume there are no α-discriminating terms and $A \cup \{[su]\} \notin \Gamma_{\mathcal{T}}$ for all normal $u \in \mathrm{EFO}_\alpha$. In particular, $[su] \notin A$ for all normal $u \in \mathrm{EFO}_\alpha$. Choose a variable $x : \alpha$ occurring free in A (or any variable $x : \alpha$ if none occurs free in A). Since

$A\cup\{[sx]\} \notin \Gamma_{\mathcal{T}}$, $A\cup\{[sx]\}$ is refutable. Using $\mathcal{F}_{\forall}$ (with the restriction), A is refutable. Contradiction.

$\mathcal{C}_{\neg\forall}$ Let $\neg\forall_\alpha s \in A \in \Gamma_{\mathcal{F}}^{\mathrm{EFO}}$. Suppose $A \cup \{\neg[sx]\} \notin \Gamma_{\mathcal{T}}$ for every variable $x : \alpha$. Let $x : \alpha$ be fresh for A. Then $A \cup \{\neg[sx]\}$ is refutable and so A can be refuted using $\mathcal{F}_{\neg\forall}$. □

Lemma 14.3 (EFO Extension Lemma)**.** *Let Γ be an abstract consistency class and $A \in \Gamma$ be an EFO branch. Then there exists an evident EFO branch E such that $A \subseteq E$. Moreover, if Γ is EFO-complete, a EFO-complete evident EFO branch E exists such that $A \subseteq E$.*

Proof. Similar to Lemma 8.2. Instead of using an enumeration of all normal formulas, we use an enumeration of all normal EFO formulas. The proof goes through when one makes some obvious modifications. □

Theorem 14.4 (EFO Completeness)**.** *Every finite EFO branch is either refutable or has a standard model.*

Proof. Follows from Lemma 14.2, Lemma 14.3 and Theorem 13.9. □

We now turn to compactness and the existence of countable models. Let $\Gamma_{\mathrm{C}}^{\mathrm{EFO}}$ be the set of all sufficiently pure EFO branches A such that every finite subset of A has a standard model.

Lemma 14.5. *$\Gamma_{\mathrm{C}}^{EFO}$ is an EFO-complete abstract consistency class.*

Proof. Similar to Lemma 10.2. □

Theorem 14.6. *Let A be a branch such that every finite subset of A has a standard model. Then A has a standard model where $\mathcal{D}\alpha$ is countable for all sorts α.*

Proof. Similar to Theorem 10.3. □

Corollary 14.7. *Let A be a satisfiable EFO branch. Then A has a standard model where $\mathcal{D}\alpha$ is countable for all sorts α.*

Proof. To apply Theorem 14.6 we only need to show every subset of A has a standard model. Let A' be a finite subset of A. Since A' is satisfiable, A' is not refutable by Proposition 14.1. By Theorem 14.4 A' has a standard model. □

15. Decidable EFO Fragments

Given the completeness result for the tableau calculus $\mathcal{F}$ (Theorem 14.4), we can show a fragment of EFO is decidable by proving $\mathcal{F}$ terminates on branches in the fragment. We will use this technique to argue decidability of three fragments:

- The *λ-free fragment*, which is EFO without λ-abstraction.
- The *pure fragment*, which consists of disequations $s \neq t$ where no name used in s and t has a type that contains o.
- The *BSR fragment (Bernays-Schönfinkel-Ramsey)*, which consists of relational first-order $\exists^*\forall^*$-formulas with equality.

Proposition 15.1 (Verification Soundness)**.** *Let A be a finite EFO branch that is not closed and cannot be extended with $\mathcal{F}$. Then A is evident and has a finite model.*

Proof. Checking A is evident is easy. The existence of a finite model follows from Theorem 13.9. □

We now have a general method for proving decidability of satisfiability within a fragment.

Proposition 15.2. *Let $\mathcal{F}$ terminate on a set Δ of finite EFO branches. Then satisfiability of the branches in Δ is decidable and every satisfiable branch in Δ has a finite model.*

Proof. Follows with Propositions 14.1 and 15.1 and Theorem 13.9. □

The decision procedure depends on the normalization operator employed with $\mathcal{F}$. A normalization operator that yields β-normal forms provides for all termination results proven in this section. Note that the tableau calculus applies the normalization operator only to applications st where s and t are both normal and t has type α (for some sort α) if it is not a variable. Hence at most one β-reduction is needed for normalization if s and t are β-normal. Moreover, no α-renaming is needed if the bound variables are chosen differently from the free variables. For clarity, we continue to work with an abstract normalization operator and state further conditions as they are needed.

15.1. **Lambda-Free Formulas.** In [15] we study lambda- and quantifier-free EFO and show that the concomitant subsystem of $\mathcal{F}$ terminates on finite branches. The result extends to lambda-free branches containing quantifiers (e.g., $\{\forall_\alpha f\}$).

Proposition 15.3 (Lambda-Free Termination)**.** *Let the normalization operator satisfy $[s] = s$ for every lambda-free EFO term s. Then $\mathcal{F}$ terminates on finite lambda-free branches.*

Proof. An application of $\mathcal{F}_{\text{FE}}$ disables a disequation $s{\neq_{\sigma\tau}}t$ and introduces new subterms as follows: a variable $x : \sigma$, two terms $sx : \tau$ and $tx : \tau$, and the formula $sx{\neq}tx$. The types of the new subterms are smaller than the type of s and t, and the new subterms introduced by the other rules always have type o or α. For each branch, consider the multiset of types $\sigma\tau$ where $s, t : \sigma\tau$ are subterms of formulas on the branch but there is no $x : \sigma$ such that $sx \neq tx$ is on the branch. By considering the multiset ordering, we see that no derivation can employ $\mathcal{F}_{\text{FE}}$ infinitely often.

Let $A \to A_1 \to A_2 \to \cdots$ be a possibly infinite derivation that issues from a finite lambda-free branch and does not employ $\mathcal{F}_{\text{FE}}$. It suffices to show that the derivation is finite. Consider the new variables $x : \alpha$ which may be introduced by the $\mathcal{F}_\forall$ and $\mathcal{F}_{\neg\forall}$ rules. For each subterm $\forall_\alpha s$ at most one new variable will be introduced by these rules. Since the branches are λ-free, no rule creates new subterms of the form $\forall_\alpha s$. Hence only finitely many new variables of type α are introduced. Let A_n be a branch in the sequence such that no new variables are introduced after this point. Let S_σ be the set of all subterms of type σ of the formulas in A_n. Let B be the union of the three finite sets S_o, $\{\neg s | s \in S_o\}$ and $\{s \neq_\sigma t | s, t \in S_\sigma\}$. Every branch A_m with $m \geq n$ can only contain members of B. Hence the derivation is finite. □

15.2. **Pure Disequations.** A type is *pure* if it does not contain o. A term is *pure* if the type of every name occurring in it (bound or unbound) is pure. An equation $s = t$ or disequation $s \neq t$ is *pure* if s and t are pure terms.

We add a new property of normalization in order to prove termination.

N5: The least relation $\succ$ on terms such that

(1) $as_1 \dots s_n \succ s_i$ if $i \in \{1, \dots, n\}$

(2) $s \succ [sx]$ if $s : \sigma\tau$ and $x : \sigma$

terminates on normal terms.

Proposition 15.4 (Pure Termination)**.** *Let the normalization operator satisfy N5. Then $\mathcal{F}$ terminates on finite branches containing only pure disequations.*

Proof. Let $A \to A_1 \to A_2 \to \cdots$ be a possibly infinite derivation that issues from a finite branch containing only pure disequations. Then no other rules but possibly $\mathcal{F}_{\text{DEC}}$ and $\mathcal{F}_{\text{FE}}$ apply and thus no A_i contains a formula that is not a pure disequation (using S5). Using N5 it follows that the derivation is finite. □

15.3. **Bernays-Schönfinkel-Ramsey Formulas.** It is well-known that the satisfiability of Bernays-Schönfinkel-Ramsey formulas (relational first-order $\exists^*\forall^*$-prenex formulas with equality) is decidable and the fragment has the finite model property [11]. We reobtain this result by showing that $\mathcal{F}$ terminates for the respective fragment. We call a type *BSR* if it is α or o or has the form $\alpha_1 \dots \alpha_n o$. We call an EFO formula s *BSR* if it satisfies two conditions:

(1) The type of every variable that occurs in s is BSR.
(2) $\forall_\alpha$ does not occur below a negation or an implication in s.

Note that every subterm of a BSR formula that has type α is a variable. For simplicity, our BSR formulas don't provide for outer existential quantification. We need one more condition for the normalization operator:

N6: If $s : \alpha o$ is BSR and $x : \alpha$, then $[sx]$ is BSR.

Proposition 15.5 (BSR Termination)**.** *Let the normalization operator satisfy N5 and N6. Then $\mathcal{F}$ terminates on finite branches containing only BSR formulas.*

Proof. Let $A \to A_1 \to A_2 \to \cdots$ be a possibly infinite derivation that issues from a finite branch containing only BSR formulas. Then $\mathcal{F}_{\neg\forall}$ and $\mathcal{F}_{\text{FE}}$ are not applicable and all A_i contain only BSR formulas (using N6). Furthermore, for each sort α used in A at most one new variable of sort α is introduced (by the restriction on $\mathcal{F}_\forall$ in $\mathcal{F}$). Since all terms of sort α are variables, there is only a finite supply. Using N5 it follows that the derivation is finite. □

16. Conclusion

In this paper we have studied a complete cut-free tableau calculus for simple type theory with primitive equality (STT). For the first-order fragment of STT (EFO) we have shown that the tableau system is complete with respect to standard models. Our development demonstrates that first-order logic can be treated naturally as a fragment of STT.

For the EFO fragment we gave an interesting restriction on instantiations. In particular, one can restrict most instantiations of sort α to be α-discriminating terms. Such a restriction can also be included in the tableau calculus for STT without sacrificing completeness. Confining instantiations to α-discriminating terms is a serious restriction since each branch has only finitely many such terms.

Automated theorem proving would be a natural application of the tableau calculi presented here. When designing a search procedure one often starts with a complete ground calculus (like our tableau calculi $\mathcal{T}$ and $\mathcal{F}$), then extends this to include metavariables to be instantiated during search, and finally proves a lifting lemma showing the tableaux with metavariables can simulate a refutation in the ground calculus. A design principle of our calculi $\mathcal{T}$ and $\mathcal{F}$ is that none of the rules look deeply into the structure of any formula on the branch. For example, consider the mating rule

$$\frac{xs_1 \dots s_n \, , \, \neg xt_1 \dots t_n}{s_1 \neq t_1 \mid \cdots \mid s_n \neq t_n} \quad n \geq 0$$

To check if this rule applies to two formulas s, t on the branch A, one only needs to check if s has a variable x at the head and if t is the negation of a formula with x at the head. When trying to prove a lifting lemma, we would need to show how the calculus with metavariables could simulate the mating rule. This may involve partially instantiating metavariables to expose the head x in the counterpart to s or the negation and the head x in the counterpart to t. On the other hand, suppose our ground calculus included a rule to close branches with a formula of the form $s \neq s$. To simulate this in the calculus with metavariables we would need to know if some instantiation for the metavariables can yield a formula of the form $s \neq s$. In the worst case this is a problem requiring full higher-order unification. We have been careful to only include rules in our calculi which will not require arbitrary instantiations of metavariables to prove a lifting lemma. Formulating such a calculus with metavariables and proving such a lifting lemma is left for future work.

References

[1] P. B. Andrews. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof.* Kluwer Academic Publishers, 2nd edition, 2002.

[2] Peter B. Andrews. Resolution in type theory. *J. Symb. Log.*, 36:414–432, 1971.

[3] Peter B. Andrews. General models and extensionality. *J. Symb. Log.*, 37:395–397, 1972.

[4] Christoph Benzmüller. *Equality and Extensionality in Automated Higher-Order Theorem Proving.* PhD thesis, Universität des Saarlandes, 1999.

[5] Christoph Benzmüller. Extensional higher-order paramodulation and RUE-resolution. In *Proc. of CADE*, volume 1632 of *LNAI*, pages 399–413. Springer, 1999.

[6] Christoph Benzmüller, Chad E. Brown, and Michael Kohlhase. Higher-order semantics and extensionality. *J. Symb. Log.*, 69:1027–1088, 2004.

[7] Christoph Benzmüller, Chad E. Brown, and Michael Kohlhase. Semantic techniques for cut-elimination in higher order logic. Technical report, Saarland University, Saarbrücken, Germany and Carnegie Mellon University, Pittsburgh, USA, 2004. Manuscript.

[8] Christoph Benzmüller, Chad E. Brown, and Michael Kohlhase. Cut-simulation and impredicativity. *Logical Methods in Computer Science*, 5(1):1–21, 2009.

[9] Christoph Benzmüller and Michael Kohlhase. Extensional higher-order resolution. In Claude Kirchner and Hélène Kirchner, editors, *Automated Deduction - CADE-15, 15th International Conference on Automated Deduction, Lindau, Germany, July 5-10, 1998, Proceedings*, number 1421 in LNCS, pages 56–71. Springer, 1998.

[10] Evert W. Beth. Semantic entailment and formal derivability. *Mededelingen der Koninklijke Nederlandse Akademie van Wetenschappen, Afdeling Letterkunde*, 18(13):309–342, 1955.

[11] Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem.* Springer, 1997.

[12] Chad E. Brown. *Set Comprehension in Church's Type Theory.* PhD thesis, Department of Mathematical Sciences, Carnegie Mellon University, 2004.

[13] Chad E. Brown. *Automated Reasoning in Higher-Order Logic: Set Comprehension and Extensionality in Church's Type Theory.* College Publications, 2007.

[14] Chad E. Brown and Gert Smolka. Extended first-order logic. In Stefan Berghofer, Tobias Nipkow, Christian Urban, and Makarius Wenzel, editors, *TPHOLs 2009*, volume 5674 of *LNCS*, pages 164–179. Springer, August 2009.

[15] Chad E. Brown and Gert Smolka. Terminating tableaux for the basic fragment of simple type theory. In M. Giese and A. Waaler, editors, *TABLEAUX 2009*, volume 5607 of *LNCS (LNAI)*, pages 138–151. Springer, 2009.

[16] Alonzo Church. A formulation of the simple theory of types. *J. Symb. Log.*, 5:56–68, 1940.

[17] Gerhard Gentzen. Untersuchungen über das natürliche Schließen I, II. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1935.

[18] Leon Henkin. Completeness in the theory of types. *J. Symb. Log.*, 15:81–91, 1950.

[19] Leon Henkin. A theory of propositional types. *Fundamenta Mathematicae*, 52:323–344, 1963.

[20] K. Jaakko J. Hintikka. Form and content in quantification theory. Two papers on symbolic logic. *Acta Philosophica Fennica*, 8:7–55, 1955.

[21] Michael Kohlhase. A unifying principle for extensional higher-order logic. Technical Report 93–153, Department of Mathematics, Carnegie Mellon University, January 1993.

[22] Michael Kohlhase. Higher-order tableaux. In Peter Baumgartner, Reiner Hähnle, and Joachim Posegga, editors, *TABLEAUX*, volume 918 of *LNCS*, pages 294–309. Springer, 1995.

[23] Reinhard Muskens. Intensional Models for the Theory of Types. *The Journal of Symbolic Logic*, 72(1):98–118, 2007.

[24] Dag Prawitz. Hauptsatz for higher order logic. *J. Symb. Log.*, 33:452–457, 1968.

[25] Raymond M. Smullyan. *First-Order Logic.* Springer, 1968.

[26] Richard Statman. Logical relations and the typed λ-calculus. *Information and Control*, 65:85–97, 1985.

[27] William W. Tait. A nonconstructive proof of Gentzen's Hauptsatz for second order predicate logic. *Bulletin of the American Math. Society*, 72(6):980–983, 1966.

[28] Moto-o Takahashi. A proof of cut-elimination theorem in simple type theory. *Journal of the Mathematical Society of Japan*, 19:399–410, 1967.

[29] Moto-o Takahashi. Simple Type Theory of Gentzen Style with the Inference of Extensionality. *Proc. Japan Acad.*, 44:43–45, 1968.

[30] Gaisi Takeuti. On a generalized logic calculus. *Japanese Journal of Mathematics*, 23:39–96, 1953. Errata: ibid, vol. 24 (1954), 149–156.

[31] Gaisi Takeuti. *Proof Theory.* Elsevier Science Publishers, 1975.